Metódy pre rozpoznávanie písaného písma

Andrej Ovečka

Bakalářská práce
2020

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav informatiky a umělé inteligence

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Andrej Ovečka**
Osobní číslo:	**A16439**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Softwarové inženýrství**
Forma studia:	**Prezenční**
Téma práce:	**Metody pro rozpoznávání psaného písma**
Téma práce anglicky:	**Hand-writing Recognition Methods**

Zásady pro vypracování

1. Proveďte rešerši aktuálního stavu v oblasti rozpoznávání psaného písma.
2. Dostupné metody vhodně kategorizujte dle principu a úspěšnosti rozpoznávání.
3. Proveďte průzkum dostupných aplikací pro rozpoznávání psaného písma.
4. Otestujte dostupné aplikace a ověřte publikované výsledky na známém nebo připraveném datasetu.
5. Zhodnoťte dosažené výsledky.

Rozsah bakalářské práce:
Rozsah příloh:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. IMPEDOVO, Sebastiano, ed., 1994. *Fundamentals in Handwriting Recognition* [online]. Berlin Heidelberg: Springer-Verlag. Nato ASI Subseries F: [vid. 2019-11-28]. ISBN 978-3-642-78648-8. Dostupné z: doi:10.1007/978-3-642-78646-4
2. LEE, Seong-Whan, 1999. *Advances in Handwriting Recognition*. Singapore; River Edge, NJ: World Scientific Pub Co Inc. ISBN 978-981-02-3715-8.
3. BEZERRA, Byron Leite Dantas, Cleber ZANCHETTIN, Alejandro H. TOSELLI a Giuseppe PIRLO, ed., 2017. *Handwriting: Recognition, Development and Analysis*. New York: Nova Science Pub Inc. ISBN 978-1-5361-1937-4.
4. BISHOP, Christopher M., 2011. *Pattern Recognition and Machine Learning*. New York: Springer. ISBN 978-0-387-31073-2.
5. FINK, Gernot A., 2014. *Markov Models for Pattern Recognition: From Theory to Applications*. 2nd ed. 2014 edition. London: Springer. ISBN 978-1-4471-6307-7.

Vedoucí bakalářské práce: **doc. Ing. Zuzana Komínková Oplatková, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce: **28. listopadu 2019**
Termín odevzdání bakalářské práce: **15. května 2020**

doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

Ve Zlíně dne 9. prosince 2019

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 7.8.2020 Andrej Ovečka, v. r.

ABSTRAKT

Témou tejto práce sú metódy pre rozpoznávanie ručne písaného písma. Cieľom je tieto metódy popísať a porovnať. Práca je rozdelená na dve časti, teoretickú a praktickú. V teoretickej časti sú popísané jednotlivé metódy, ktoré sa pri rozpoznávaní písma používajú a datasety s výsledkami opísaných klasifikátorov. Praktická časť práce sa venuje prieskumu a testovaniu konkrétnych aplikácií. Záver práce je zameraný na zhrnutie zistených poznatkov.

Kľúčové slová: rozpoznávanie ručne písaného písma, k-najbližších susedov, metóda podporných vektorov, Bayesov naivný klasifikátor, viacvrstvové perceptróny, LSTM, konvolučné neurónové siete, dataset

ABSTRACT

The topic of this thesis is handwriting recognition methods. The aim is to describe and compare these methods. The thesis is divided into two parts, theoretical and practical. The theoretical part describes various methods used in hand-writing recognition and datasets with the results of the described classifiers. The practical part of this thesis deals with the research and testing of specific applications. The conclusion of the thesis is focused on a summary of the findings.

Keywords: hand-writing recognition, k-Nearest Neighbors, support vectore machines, Naïve Bayes classifier, multi-layer perceptrons, LSTM, convolutional neural network, dataset

Moje poďakovanie patrí vedúcej mojej bakalárskej práce, váženej pani doc. Ing. Zuzane Komínkovej Oplatkovej Ph.D., za venovaný čas a spoluprácu pri konzultáciách tejto práce. Taktiež by som chcel poďakovať svojej rodine a priateľke, za to, že ma podporovali počas celého štúdia.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD **9**
I TEORETICKÁ ČASŤ **11**
1 KLASIFIKÁCIA **12**
1.1 K-NEAREST NEIGHBORS 12
1.2 SUPPORT VECTOR MACHINES 14
1.3 VIACVRSTVOVÉ PERCEPTRÓNY 16
1.4 KONVOLUČNÉ NEURÓNOVÉ SIETE 17
1.5 BAYESOV NAIVNÝ KLASIFIKÁTOR 19
1.6 LONG SHORT-TERM MEMORY 20
2 PREPROCESSING **22**
2.1 ODSTRÁNENIE ŠUMU 22
2.1.1 Lineárne filtre 22
2.1.2 Mediánová filtrácia 22
2.2 DETEKCIA A KOREKCIA ZOŠIKMENIA 23
2.3 NORMALIZÁCIA OBRAZU 24
2.4 ROZMAZANIE A VZORKOVANIE 25
3 SEGMENTÁCIA TEXTU **26**
3.1 SEGMENTÁCIA RIADKOV 26
3.2 SEGMENTÁCIA SLOV 27
3.3 SEGMENTÁCIA ZNAKOV 27
3.3.1 Explicitná segmentácia 27
3.3.2 Implicitná segmentácia 29
3.3.3 Holistická segmentácia 29
4 EXTRAKCIA PRÍZNAKOV **31**
4.1 ŠTATISTICKÉ PRÍZNAKY 31
4.1.1 Zoning 31
4.1.2 Projekčné histogramy 32
4.1.3 Profily 32
4.1.4 Kríženia a vzdialenosti 33
4.2 ŠTRUKTURÁLNE PRÍZNAKY 33
4.3 GLOBÁLNE TRANSFORMÁCIE – MOMENTY 34
5 POST-PROCESSING **35**
6 DATASETY **36**
6.1 MNIST 37
6.2 EMNIST 38

6.3 Arabské číslice 38
6.4 Arabské znaky 39
6.5 Oriya číslice 39
6.6 Bengali Handwritten Digits Database 40
6.7 Bangla číslice 40
6.8 Bangla znaky 41
6.9 Devanagari znaky 41
6.10 Gurmukni znaky 42
II PRAKTICKÁ ČASŤ 43
7 APLIKÁCIE PRE ROZPOZNÁVANIE PÍSANÉHO PÍSMA 44
7.1 Handwritten Digit Recognition using Deep Learning 45
7.1.1 k-NN 45
7.1.2 Support Vector Machines 46
7.2 MNIST_classification 47
7.3 RNN and LSTM for Image Classifications 48
7.4 MLP-for-MNIST-dataset 49
7.5 Zhodnotenie dosiahnutých výsledkov 50
ZÁVER 52
ZOZNAM POUŽITEJ LITERATÚRY 54
ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK 64
ZOZNAM OBRÁZKOV 65
ZOZNAM TABULIEK 66
ZOZNAM PRÍLOH 67

ÚVOD

Predávanie informácií vo forme rukou napísaného textu prežilo stovky rokov. Prvé pokusy o zostrojenie písacieho stroja siahajú do prvej polovice 18. storočia. Dnes sú už nahradené počítačmi a klávesnicou.

Rozvoj informatiky a informačných technológií rýchlo napreduje. To, čo sa pred pár desaťročiami zdalo nemožné, je v dnešnej dobe samozrejmosťou a stáva sa neoddeliteľnou súčasťou nášho každodenného života.

V súčasnosti digitalizácia zasahuje do každého odvetvia. Príkladom sú podnikové informačné systémy, elektronické zbierky zákonov a predpisov, elektronizácia zdravotníctva, elektronické archívy knižníc, ale aj budovanie elektronických archívov historických knižníc. Dôvodom digitalizácie je napríklad šetrenie miesta, zníženie nákladov na archiváciu, ochrana údajov, alebo zdieľanie textu. Zrýchľuje a zjednodušuje sa vyhľadávanie v textoch, dajú sa efektívne analyzovať, kategorizovať, spájať a porovnávať s inými údajmi a dokumentami.

Rozpoznávanie rukou písaného textu je schopnosť počítača prijímať a spracovať rukou písaný vstup, v podobe papierového dokumentu, fotografie, z dotykovej obrazovky, alebo iného zariadenia. Vstupný obrázok rukou písaného textu môže byť snímaný off-line, kedy je text na papier napísaný perom a dáta sa získavajú až po ich napísaní, alebo je snímaný on-line. V prípade on-line rozpoznávania sa dáta získavajú pomocou špeciálneho pera a dotykovej obrazovky, a sú snímané po celú dobu, kedy je pero v kontakte s obrazovkou.

Cieľom mojej práce je opísať a porovnať metódy pre rozpoznávanie rukou písaného písma a vybrať vhodný klasifikátor pre rozpoznávanie písma kurent, ktorý sa používal v minulosti napríklad v matrikách. Práca sa skladá z dvoch častí, teoretickej a praktickej. Teoretická časť sa zaoberá rešeršou techník a metód pre rozpoznávanie písaného písma. V prvej kapitole sú popísané klasifikátory, ktoré sú najdôležitejšou časťou aplikácií pre rozpoznávanie. V ďalších kapitolách teoretickej časti opisujem metódy, ktoré sa používajú pre prípravu textu pred samotnou klasifikáciou. Konkrétne to sú to metódy segmentácie textu, predspracovania, extrakcie príznakov a post-processing. Posledná kapitola teoretickej časti opisuje datasety, na ktorých sú klasifikátory trénované a následne aj testované. Praktická časť mojej práce je zameraná na prieskum a otestovanie konkrétnych aplikácii pre rozpoznávanie písaného písma. Prvá kapitola praktickej časti obsahuje prieskum

a testovanie aplikácii na datasete MNIST a druhá kapitola obsahuje zhrnutie dosiahnutých výsledkov.

I. TEORETICKÁ ČASŤ

1 KLASIFIKÁCIA

Klasifikácia je proces kategorizácie daného súboru údajov do tried. Môže sa vykonávať na štruktúrovaných aj neštruktúrovaných údajoch. Triedy sa často označujú aj ako cieľ, štítok alebo kategórie [59].

Klafikátor je algoritmus, ktorý sa používa na mapovanie vstupných údajov do konkrétnej triedy alebo kategórie [59]. V nasledujúcich podkapitolách sú popísané klasifikátory:

- k-Nearest Neighbor (k-Najbližších Susedov, k-NN)
- Support Vector Machines (SVM),
- Viacvrstvové perceptróny (Multi-layer Perceptrons, MLP)
- Konvolučné neurónové siete (CNN),
- Bayesov naivný klasifikátor (NB),
- Long short-term memory (LSTM).

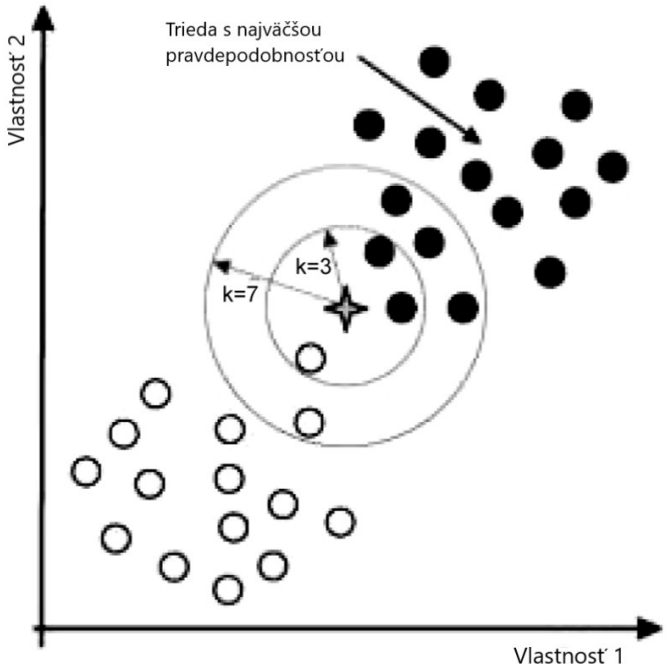
Pre zvýšenie presnosti klasifikácie sa používajú rôzne techniky predspracovania obrazu a extrakcie príznakov, ktoré odstraňujú, alebo redukujú nežiaduce informácie z obrázka. Tieto techniky sú popísané v kapitolách 2 a 4.

1.1 k-Nearest Neighbors

Metóda najbližších susedov (k-NN) je jedným z najpoužívanejších a najjednoduchších algoritmov v strojovom učení, bez toho, aby vyžadovala väčšie nároky na svojho autora. Napriek svojej jednoduchosti dosahuje uspokojivé výsledky, ak sú splnené určité požiadavky. Základnou myšlienkou je, že podobné príklady sa vyskytujú „blízko“ pri sebe, čo sa týka ich vlastností. Inými slovami, ak vzorka patrí do určitej triedy, očakáva sa, že hodnoty vlastností inej vzorky, ktorá patrí do tej istej triedy, sa príliš neodlišujú od hodnôt prvej vzorky. Takže vzdialenosť medzi týmito inštanciami by mala byť malá [1].

Preto je k-NN neparametrická metóda, ktorá funguje nasledovne. Uchovávame všetky dostupné vzorky, ktoré zodpovedajú tréningovým dátam. Tie sa skladajú z ich vlastností a značiek. Potom porovnávame vstupnú vzorku so všetkými tréningovými dátami a priradíme ju k triede s väčšinou k najbližších vzoriek z tréningovej sady. Parameter k je definovaný užívateľom. Obrázok číslo 1 zobrazuje princíp funkcionality k-NN. V tomto prípade je vstup označený ako hviezdička. Ďalej máme dve triedy, označené ako prázdne

a plné kruhy. V oboch prípadoch, zobrazených na obrázku 1, k = 3 a k = 7, metóda predpovedá, že vstupná vzorka patrí do triedy označenej plnými kruhmi [1].



Obrázok 1: Princíp fungovania k-NN [1]

Je teda možné pozorovať, že parameter *k* hrá kľúčovú úlohu vo výsledkoch k-NN. Nie je ťažké si všimnúť, že v závislosti od tejto hodnoty môže algoritmus zmeniť svoju predpoveď. Z toho vyplýva, najlepšia hodnota pre tento parameter závisí od problému a konkrétne od údajov. Bežnou praxou pri odhadovaní hodnoty *k* je zmeniť hodnotu z jednotky na druhú odmocninu počtu tréningových vzoriek a vybrať hodnotu, ktorá dosiahla najlepšie výsledky [1].

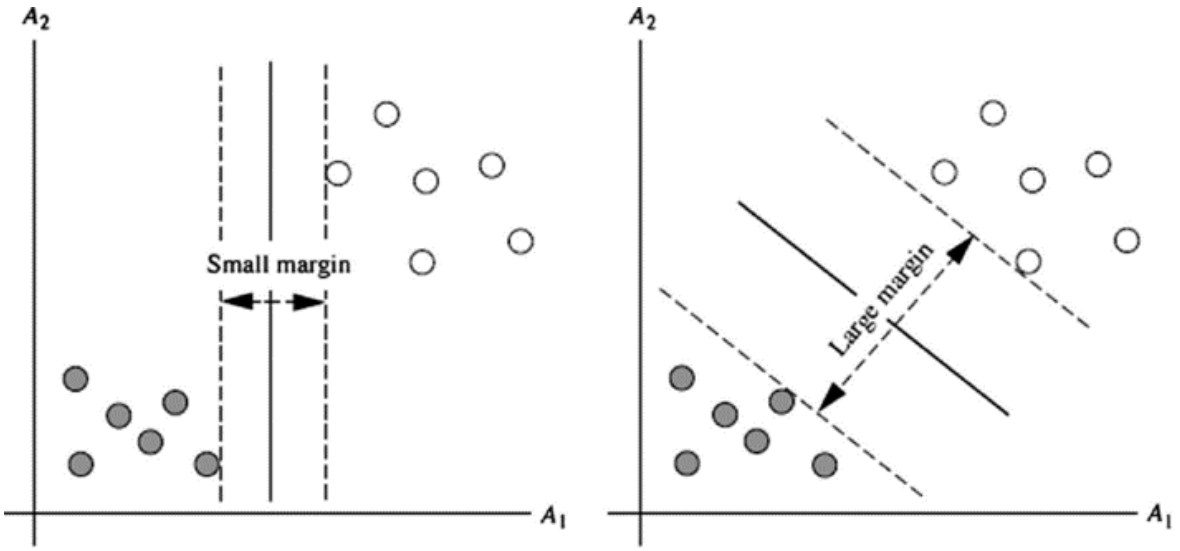
Určenie miery priblíženia závisí na meranej vzdialenosti. Najpoužívanejšou metrikou vzdialenosti v k-NN je euklidovská vzdialenosť, hoci v literatúre je možné nájsť ďalšie, ako napríklad city-block, Manhattan, Mahalanobis, alebo Minkowski [1].

Nedostatky k-NN sú hlavne závislosť na štruktúre dát a preťaženie spracovávania pri narastajúcom množstve tréningových sád [1]. Tréningová sada je časť datasetu, na ktorej prebieha učenie algoritmu. Čo je to dataset je popísané v kapitole 6. Keďže je algoritmus založený na porovnávaní vstupov a tréningových dát, čím väčšia je tréningová sada, tým väčšie je množstvo operácií, a teda čas spracovania. Aby sa tomuto problému predišlo, využíva sa postup skrátenia, ktorý sa zameriava na zníženie počtu tréningových príkladov, zvyčajne založených na určitej podobnosti. V tejto súvislosti je známe, že niekoľko podobných príkladov pravdepodobne nepomôže rozlíšiť údaje. Niektoré z týchto príkladov

by sa preto mohli vylúčiť bez straty zovšeobecnenia, čo by pomohlo zlepšiť výkonnosť metódy [1].

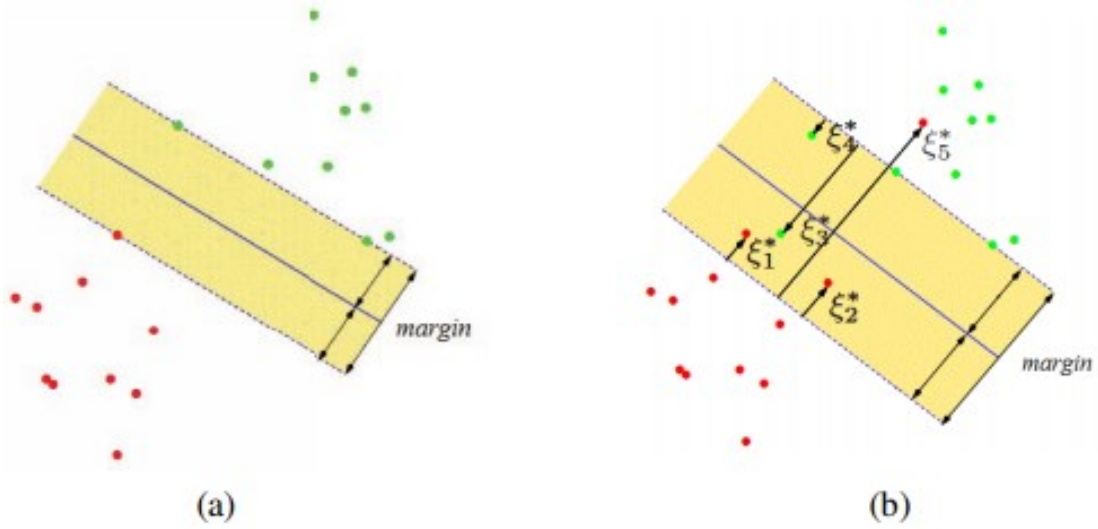
1.2 Support Vector Machines

Support Vector Machines (SVM), alebo metóda podporných vektorov, je binárna metóda strojového učenia založená na štatistickej teórii učenia s niektorými vysoko elegantnými vlastnosťami. Hlavná myšlienka sa dá zhrnúť nasledovne: pri tréningovej vzorke SVM vytvára nadrovinu, ako rozhodovaciu plochu takým spôsobom, aby sa maximalizovala miera oddelenia medzi pozitívnymi a negatívnymi príkladmi. Táto vlastnosť je v kontraste s inými technikami učenia, ktoré využívajú akúkoľvek separačnú plochu, napríklad viacvrstvové perceptróny alebo funkcia radiálnej bázy (RBF). Rozpätie oddelenia je definované ako menšia vzdialenosť medzi tréningovými vzorkami a rozhodovacou plochou, ktorá sa v tejto situácii môže považovať za optimálnu nadrovinu. Obrázok 2 zobrazuje rozdiel medzi náhodnou rozhodovacou plochou s menším odstupom a optimálnou nadrovinou, ktorá má väčší odstup separácie [1].



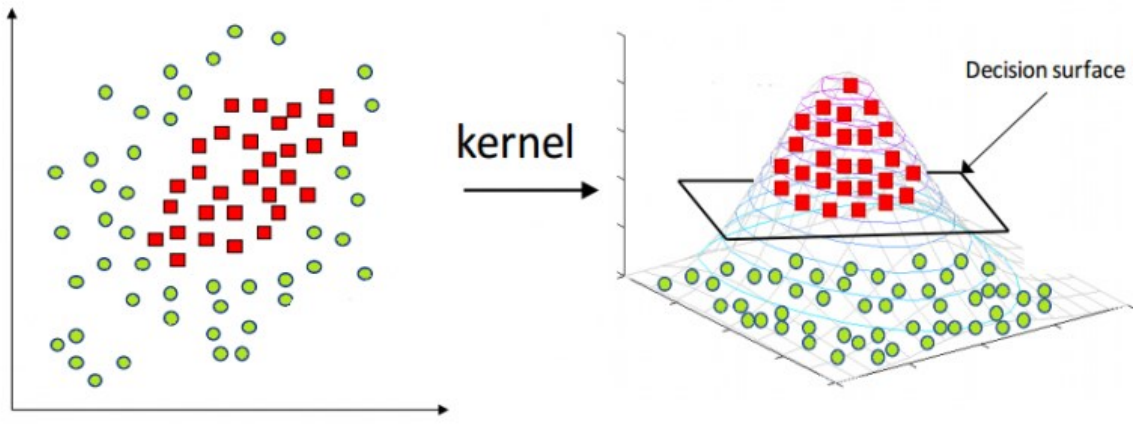
Obrázok 2: Rozhodovacie plochy [55]

Základný postup na určenie optimálnej nadroviny umožňuje len klasifikáciu lineárne separovateľných údajov. S cieľom spracovať neoddeliteľné údaje sa zavádza pojem zašumené dáta (soft margin). V takom prípade sú klasifikačné chyby pri tréningu povolené, aby bol odstup širší, čo má tendenciu zvyšovať generalizačnú silu klasifikátora . Obrázok 3 zobrazuje túto situáciu a zobrazuje tak lineárne oddeliteľné údaje (a), ako aj neoddeliteľné údaje (b), kde body označené ako ξ_3 a ξ_5 sú na nesprávnej strane rozhodovacej plochy [1].



Obrázok 3: Klasifikácia podporných vektorov: (a) lineárne oddeliteľné dáta, (b) lineárne neoddeliteľné dáta [1]

Napriek tomu, že je zašumenie dát veľmi užitočné, nedokáže poskytnúť SVM požadované klasifikačné schopnosti. Vyplýva to zo skutočností, že nadroviny nie sú vždy dostatočné na oddelenie vstupných údajov. Existujú situácie, v ktorých by bola vhodnejšia nelineárna rovina. Táto konverzia, alebo mapovanie je dosiahnuté pomocou jadrovej transformácie (Kernel Trick). Myšlienkou je, že nelineárne oddeliteľné vstupné dáta sa môžu stať lineárne oddeliteľnými v inej nadrovine, v ktorej bude možné definovať nadrovinu, ktorá diskriminuje dané dáta. Obrázok 4 predstavuje takúto transformáciu [1].



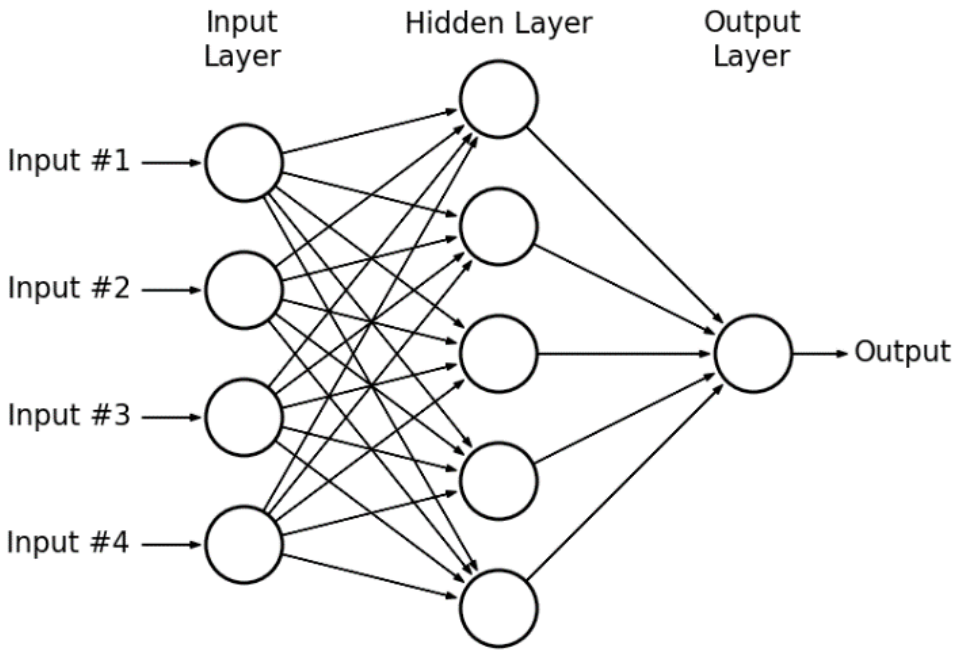
Obrázok 4: Kernel mapping: nelineárne oddeliteľné dáta a lineárne oddeliteľné dáta [2]

Rovnako ako iné klasifikátori, aj SVM má parametre, ktoré definujú jeho výkon pri určitom probléme. Hlavnými parametrami sú jadrová funkcia a jej vnútorné parametre. Jadrová funkcia môže byť polynóm, Gaussova krivka, alebo RBF a regularizačná konštanta použitá

na zašumenie dát. SVM sú citlivejšie na zmenu svojich parametrov, čo sa považuje za nedostatok metódy [1].

1.3 Viacvrstvové Perceptróny

Viacvrstvové perceptróny (MLP z anglického Multi-layer Perceptron) sú sieťami s priamym posuvom, ktoré majú jednu, alebo viac skrytých vrstiev, obvykle zložených z neurónov sigmiodálnej aktivácie. Zaujímavé vlastnosti MLP sú, že s jednou skrytou vrstvou môže aproximovať akúkoľvek spojitú funkciu, zatiaľ čo použitie dvoch skrytých vrstiev umožňuje aproximáciu akejkoľvek funkcie. Obrázok 5 zobrazuje schematický pohľad na MLP sieť so štyrmi uzlami vo vstupnej vrstve, jednou skrytou vrstvou so štyrmi uzlami a výstupnou vrstvou s jedným uzlom [1].



Obrázok 5: Príklad viacvrstového perceptrónu [3]

Spočiatku boli neurónové siete tvorené jednou vrstvou. Ich výcvik bol priamočiary, pretože výstup je priamo pozorovateľný, a preto sa môže použiť na nastavenie vodiacich váh. Faktom je, že jednovrstvové siete dokázali vyriešiť iba lineárne oddeliteľné problémy. Riešenie tohto obmedzenia sa objavilo s opisom algoritmu backprogpagation [1].

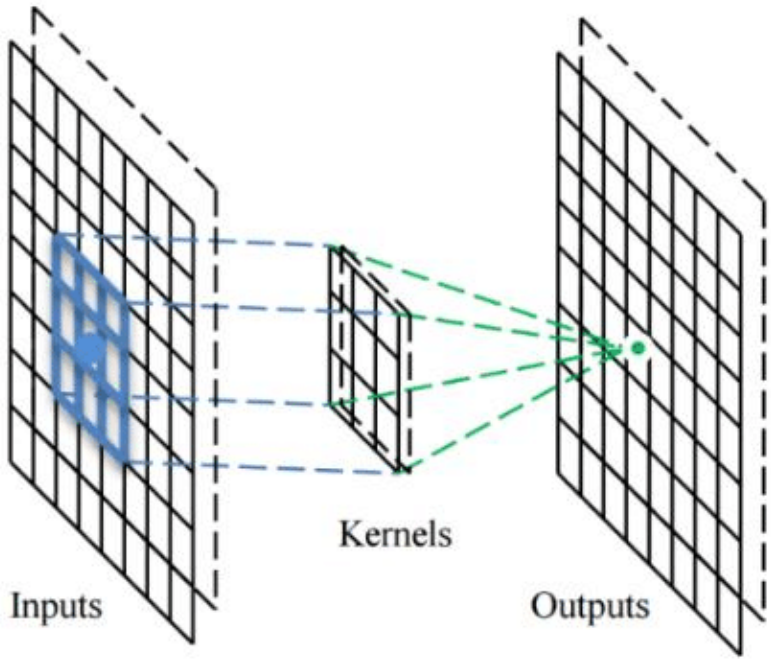
Základnou myšlienkou algoritmu je použitie zostupného gradientu na výpočet chýb skrytých vrstiev podľa odhadu efektu spôsobeného nimi, cez chybu výstupnej vrstvy. Takto sa vypočíta výstupná chyba, ktorá sa vráti späť do skrytých vrstiev, čo umožní, aby sa váhy aktualizovali úmerne k hodnotám spojení medzi vrstvami. Kvôli použitiu gradientu musia

byť aktivačné funkcie diferencovateľné. Toto opodstatňuje použitie sigmoidných funkcií, pretože sú diferencovateľnou aproximáciou ku krokovej funkcii. Učenie prebieha v dvoch fázach. V doprednej fáze je signál šírený sieťou, až kým nedosiahne výstup. Pri spätnej fáze sa chyba získa porovnaním výstupu siete s požadovanou odozvou. Táto výsledná chyba sa šíri z výstupu na vstup (smerom dozadu), vrstvu po vrstve, čo umožňuje nastavenie váh siete [1].

1.4 Konvolučné neurónové siete

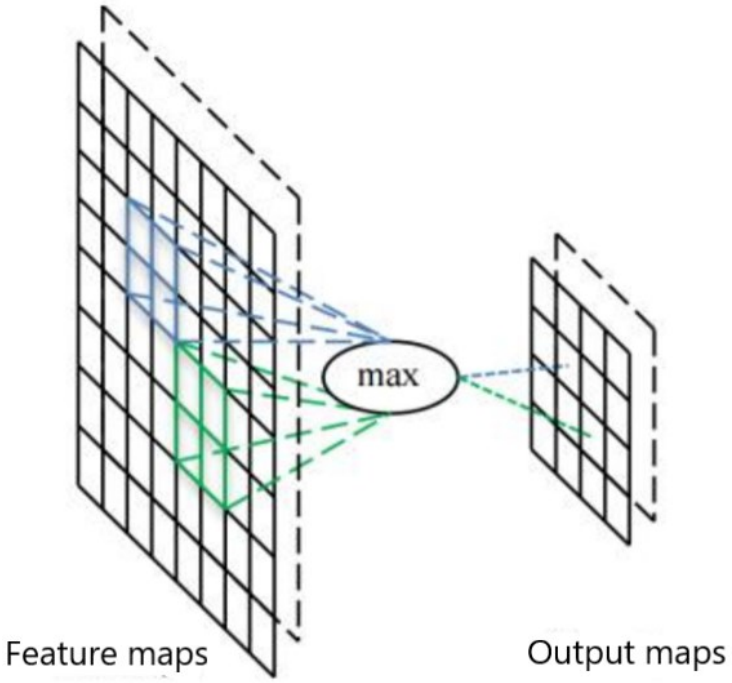
Myšlienka konvolučných sietí (CNN z angl. Convolutional Neural Network) bola vyvinutá skupinou neurovedcov a programátorom Yannom LeCunom [1]. CNN rovnako ako aj MLP patrí do skupiny neurónových sietí. Zatiaľ čo viacvrstvové perceptróny sa radia medzi plytké siete, konvolučné neurónové siete sa radia medzi metódy hlbokého učenia. Rovnako ako mozog, konvolučná sieť odovzdáva informácie cez niekoľko hierarchických vrstiev predbežného spracovania, kde každá vrstva vykonáva špecifickú úlohu. CNN tvoria tri hlavné typy vrstviev [1].

Obrázok číslo 6 zobrazuje konvolučnú vrstvu, ktorá je skonštruovaná tak, že graficky kóduje niektoré príznaky, ako napríklad prítomnosť šikmej čiary pod určitým uhlom. Ďalšia vrstva, vytvorená v dôsledku konvolučných operácií s váhovou maticou, ukazuje prítomnosť šikmej línie v ošetrovacej vrstve a jej polohu, čím vytvára tzv. príznakové mapy. Konvolučná neurónová sieť nemá jednu množinu takýchto váh, ale má ich celú škálu, kódujúcu všetky druhy čiar a oblúkov z rôznych uhlov. Takéto konvolučné jadrá nie sú nachystané dopredu, ale sú tvorené samostatne prostredníctvom trénovania siete klasickým spôsobom spätnej propagácie (backpropagation) [4].



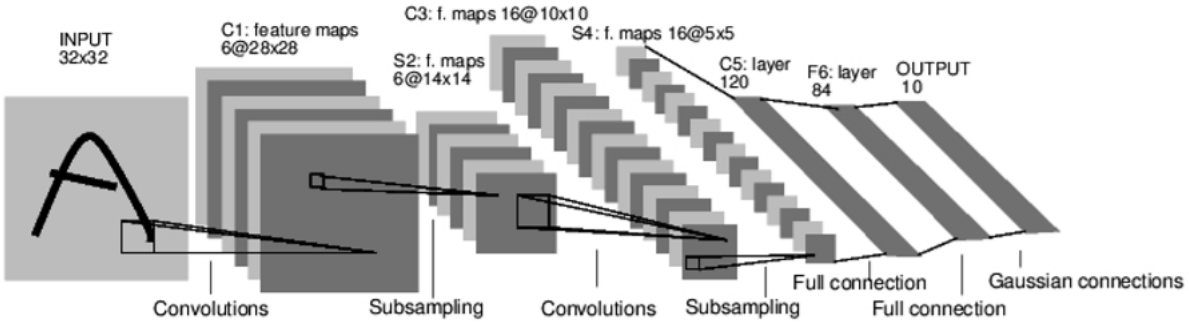
Obrázok 6: Konvolučná vrstva [4]

Subsamplingová (pooling) vrstva vykonáva zmenšenie rozmerov vytvorených príznakových máp. Subsampling sa týka procesu, keď z niekoľkých susediacich neurónov príznakovej mapy sa vyberie ten, ktorý má maximálnu výstupnú hodnotu (max-pooling), prípadne sa spočíta ich priemer (average-pooling). Neskôr tvorí jeden neurón z ďalšej príznakovej mapy so zmenšenými rozmermi. Vďaka tejto operácii sa sieť okrem zrýchlenia ďalších výpočtov stáva invariantnejšou k mierke vstupného obrazu [4].



Obrázok 7: Subsamplingová (pooling) vrstva [4]

Posledná vrstva je obvykle plne prepojená vrstva, kde sú príznakové mapy zreťazené do jedného vektora a predané do vrstvy, ktorá vráti pravdepodobnosti inštancie patriacej do predtým trénovanej triedy. Tréning konvolučnej siete sa vykonáva pomocou algoritmu backpropagation. V tomto algoritme sa výsledok klasifikácie siete porovná s labelom príkladnej triedy. Potom sa chyba klasifikácie spätne propaguje smerom k predchádzajúcim vrstvám, takže sa aktualizujú ich váhy [1].



Obrázok 8: Konvolučná neurónová sieť [1]

1.5 Bayesov naivný klasifikátor

Bayesov naivný klasifikátor (NB z angl. Naive Bayes) je pravdepodobnostný klasifikátor, ktorý uplatňuje Bayesovu teorému so silným (naivným) predpokladom. Predpokladá sa, že vlastnosti popisujúce objekty, ktoré sa majú klasifikovať, sú navzájom štatisticky nezávislé. Takýto štatistický model sa v literatúre často nazýva „independent feature model“.

Vzhľadom na priestor *X*, v ktorom sú klasifikované objekty reprezentované premennými f_1, ..., f_m, je NB založená na Bayesovej vete (1):

$$p(c|f_1, \dots, f_m) = \frac{p(c)p(f_1, \dots, f_m|c)}{p(f_1, \dots, f_x)} \tag{1}$$

kde *c* je premenná triedy. Obyčajne sa menovateľ vyššie uvedenej rovnice neberie do úvahy, pretože je konštantný vzhľadom na triednu premennú *c.*

Podľa základného predpokladu NB, menovite štatistickej nezávislosti prvkov, možno uvedené rovnicu prepísať nasledujúcim spôsobom (2):

$$p(c|f_1, \dots, f_m) = \frac{1}{Z}p(c)\prod_{i=1}^{m} p(f_i|c) \tag{2}$$

Kde $Z = p(f_1, \dots, f_m)$ je mierkový faktor nezávislý od *c.* Podľa odhadu maximálnej pravdepodobnosti sa parametre modelu odhadujú z tréningového súboru údajov (povedzme

D). Predchádzajúca trieda sa odhaduje takto: ak n_k je počet vzoriek v *D* patriacich do triedy *k* a n_D je celkový počet vzoriek v *D*, potom $P(c = k) = {n_k}/{n_D}$. Pokiaľ ide o odhad parametrov distribúcie prvkov, je možné postupovať podľa dvoch rôznych prístupov v závislosti od toho, či znaky predpokladajú kontinuálne alebo diskrétne hodnoty [20], [21].

Po odhadnutí vyššie uvedených parametrov klasifikuje NB neznámu vzorku *x* predstavovanú znakovým vektorom $x = (f_1, \dots, f_m)$ podľa rozhodovacieho pravidla *maximum a posteriori* (3):

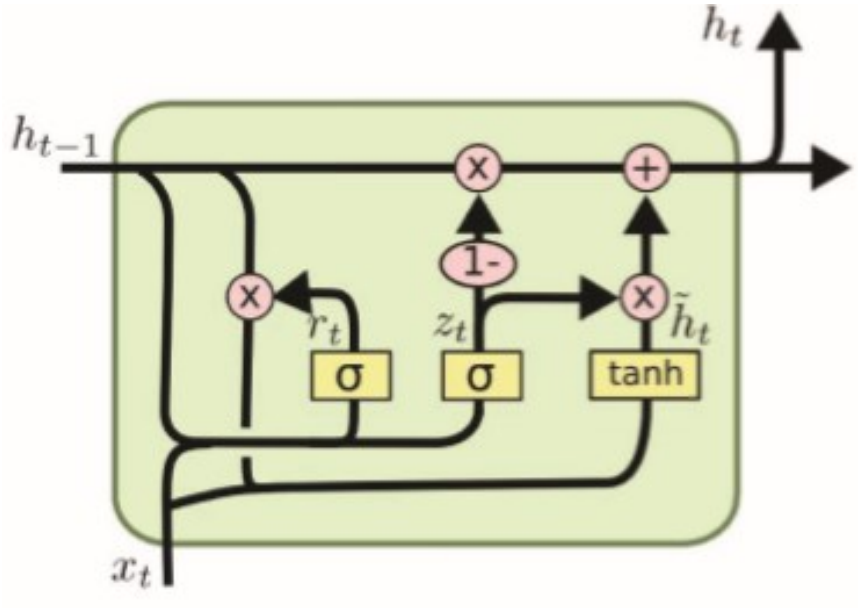
$$e(x) = \arg\max_k p(c = k) \prod_{i=1}^{m} p(f_i | c = k) \tag{3}$$

Napriek skutočnosti, že predpoklady nezávislosti môžu byť v niektorých prípadoch nepresné, NB má niekoľko vlastností, vďaka ktorým je veľmi efektívna v praktických aplikáciách. Napríklad faktorizácia distribúcie podmienených znakov triedy znamená, že každú distribúciu je možné odhadnúť nezávisle. To pomáha predchádzať problému prekliatia dimenzionality, t.j. potreba veľmi veľkých súborov údajov, ktorých počet vzoriek sa exponenciálne prispôsobuje počtu funkcií [21].

1.6 Long short-term memory

Rekurentné neurónové siete sú skupinou sietí používaných na spracovanie sekvenčných dát. Pre tento typ dát sú sieťové údaje zdieľané medzi rôznymi časovými obdobiami. Rekurentné neurónové siete by teoreticky mali dobre zvládnuť problémy so sekvenciami akejkoľvek veľkosti, od veľmi dlhých po najkratšie. V praxi sa to však nedeje. Tento problém musí byť spôsobený miznúcim gradientom [1].

Long short-term memory (LSTM) je model rekurentnej neurónovej siete, ktorý zavádza novú štruktúru nazývanú "memory cell" na riešenie problému úbytku gradientu. Pretože ide o rekurentnú sieť, jej architektúra je veľmi podobná ostatným modelom tejto skupiny, pričom sa odlišuje presne použitím štruktúry pamäťových buniek [1].



Obrázok 9: Vizualizácia pamäťovej bunky [1]

Pamäťová bunka sa skladá z prvkov, ktoré umožňujú prístup a regulujú zdieľanie informácií medzi neurónmi pri klasifikácii sekvencie. Sú to komponenty, vstupná brána, neurón so „self-recurrent“ spojením, pamäťová a výstupná brána. Self-recurrent spojenie zaisťuje, že sa ignoruje všetka interferencia zvonku, čo umožňuje, aby sa stav pamäťovej bunky udržiaval neustále medzi jedným časovým obdobím a druhým. Vstupná brána riadi interakcie medzi pamäťovými bunkami a prostredím a môže blokovať alebo umožniť vstupným signálom zmeniť stav pamäťovej bunky. Výstupná brána môže umožniť, alebo zabrániť, aby stav pamäťovej bunky mal vplyv na iné neuróny. Nakoniec pamäťová brána riadi rekurzívne retro pripojenie pamäťovej bunky, čo umožňuje bunke zapamätať si, alebo zabudnúť jej predchádzajúcu fázu, keď je to potrebné [1].

2 PREPROCESSING

Preprocessing je proces vylepšenia obrazu, ktorý by sa mal použiť na ďalšie spracovanie. Predspracovanie je hlavným krokom v systéme rozpoznávania rukopisu [5]. Cieľom predpracovania je zbaviť sa irelevantných informácií, ktoré môžu negatívne ovplyvniť rozpoznávanie [6].

2.1 Odstránenie šumu

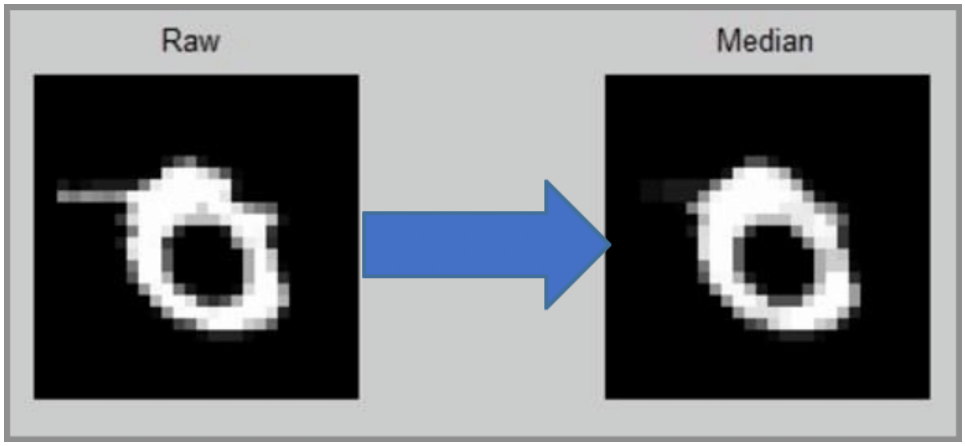
Hlavným cieľom odstraňovania šumu je odstrániť všetky nežiaduce bitové obrazce, ktoré nemajú na výstupe žiadny význam. Techniky redukcie hluku sú filtrovanie, morfologické operácie a modelovanie hluku. Filtre môžu byť navrhnuté na vyhladenie, zaostrenie, prahovanie, odstránenie mierne textúrovaného pozadia a procesu úpravy kontrastu. Rôzne morfologické operácie môžu byť navrhnuté tak, aby spájali prerušené ťahy, rozložili spojené ťahy, vyhladili kontúry, orezali nechcené body, zúžili znaky a vytiahli hranice [7].

2.1.1 Lineárne filtre

Autori v [7] preukázali, že na obrázkoch sivej úrovne najlepšie eliminuje šum izolovaného pixelu filter s nízkym priechodom, ako aj priemerné alebo Gaussové rozmazanie [7].

2.1.2 Mediánová filtrácia

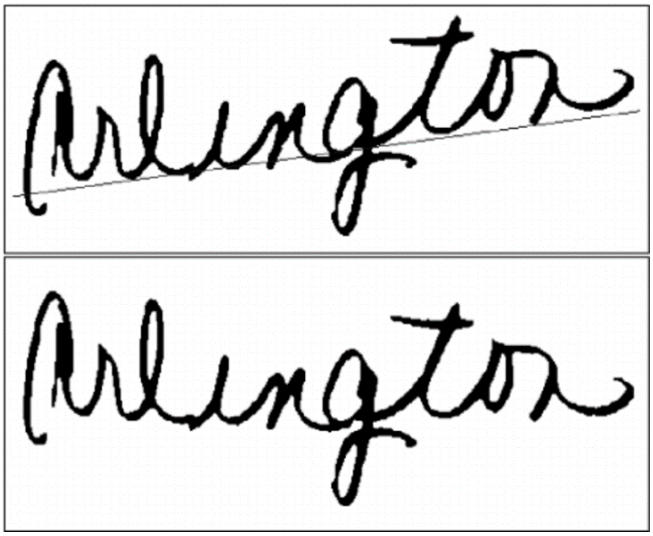
Mediánový filter sa používa ako nelineárny filter. Na odstránenie šumu izolovaného pixelu na obrázkoch sivej úrovne sa podľa autorov [7] ako najlepší osvedčil mediánový filter. Je to dolnopriepustný filter. Mediánový filter sníma oblasť obrázka (3x3, 5x5, 7x7 atď.). Vytriedi všetky hodnoty pixelov v tejto oblasti a nahradí stredný pixel strednou hodnotou. V prípade, že skúmaná oblasť obsahuje párny počet pixelov, použije sa priemer z dvoch stredných hodnôt pixelov. Mediánový filter je účinný pri odstraňovaní impulzného šumu, napríklad „salt and pepper noise“, čo je náhodný výskyt čiernobielych pixelov. Šum prítomný na obrázku sa odstráni použitím mediánového filtra. Mediánový filter je účinnejší ako konvolúcia, ak je cieľom súčasne znížiť hluk a zachovať hrany [7].



Obrázok 10: Nespracovaný obrázok (vľavo) a obrázok po použití Mediánovej filtrácie (vpravo) [56]

2.2 Detekcia a korekcia zošikmenia

Z dôvodu možnosti rotácie vstupného obrazu a citlivosti mnohých metód analýzy obrazu dokumentu na rotáciu obrázka, by sa mala korekcia skreslenia dokumentu opraviť. Techniky detekcie zošikmenia (skew detection) možno rozdeliť do nasledujúcich skupín: analýza projekčného profilu, Houghova transformácia, pripojené komponenty a zoskupovanie a korelácia medzi technikami čiar. Hull a Taylor [62] skúmali dvadsaťpäť rôznych metód detekcie skreslenia obrazu dokumentu. Metódy zahŕňajú postupy založené na analýze Houghovej transformácie, projekčných profiloch, feature point distribution and orientation-sensitive feature analysis. Prieskum dospel k záveru, že väčšina techník uviedla rozsah 0,1 rádu presnosti, čo svedčí o silnej potrebe ďalšej práce v tejto oblasti, aby sa pomohlo poukázať na silné a slabé stránky jednotlivých algoritmov (Hull & Taylor, 1998) [62]. Okrem toho sa objavujú nové techniky pre konkrétne aplikácie, ako napríklad metóda Al-Shatnawi a Omar, ktorá je založená na ťažisku pri spracovávaní obrázkov arabských dokumentov (Al-Shatnawi & Omar, 2009) [61]. Preto výber použitia techniky detekcie / korekcie zošikmenia závisí od aplikácie a typu použitých obrazov [8].



Obrázok 11: Ručne písané slovo pred a po korekcii zošikmenia [57]

2.3 Normalizácia obrazu

Normalizácia je lineárny proces. Ak je rozsah intenzity obrazu 50 až 180 a požadovaný rozsah je 0 až 255, proces vyžaduje odčítanie 50 od každej z intenzity pixelu, čím sa dosiahne rozsah 0 až 130. Potom sa každá intenzita pixelov vynásobí 255/130, čím sa dosiahne rozsah 0 až 255. Automatická normalizácia v softvéri na spracovanie obrazu sa zvyčajne normalizuje s úplným dynamickým rozsahom číselného systému špecifikovaného vo formáte obrazového súboru. Normalizačný proces vytvorí oblasti, ktoré majú rovnaké konštantné rozmery. Cieľom normalizačných metód je odstrániť všetky typy variácií počas zápisu a získajú sa štandardizované údaje. Napríklad normalizácia veľkosti sa používa na prispôsobenie veľkosti znakov určitému štandardu. Metódy rozpoznávania znakov môžu používať normalizáciu horizontálnej aj vertikálnej veľkosti. Šírka a výška znaku sa označujú pomocou W_1 a H_1, šírka a výška normalizovaného znaku sa označujú W_2 a H_2, a veľkosť štandardnej roviny je označená L. Veľkosť štandardnej roviny sa zvyčajne považuje za štvorec a jej veľkosť je medzi ostatnými zvyčajne 32 × 32 alebo 64 × 64. Pomer strán R_1 (4) a R_2 (5) pôvodného a normalizovaného charakteru je:

$$R_1 = \frac{\min(W_1, H_1)}{\max(W_1, H_1)} \tag{4}$$

$$R_2 = \frac{\min(W_2, H_2)}{\max(W_2, H_2)} \tag{5}$$

Zvažujú sa v rozmedzí [0,1] [7].

2.4 Rozmazanie a vzorkovanie

Každá smerová rovina s rovnakou veľkosťou ako normalizovaný obraz sa musí zmenšiť, aby sa získali hodnoty funkcií so strednou rozmernosťou. Jednoduchým spôsobom je rozdeliť smerovú rovinu do niekoľkých blokových zón a celkovú, alebo priemernú hodnotu každej zóny považovať za hodnotu prvku. Oddelenie zón s premenlivou veľkosťou bolo navrhnuté na prekonanie nerovnomerného rozdelenia hustoty ťahu, ale nie je potrebné pre nelineárnu alebo pseudo 2D normalizáciu. Prekrývajúce sa bloky zmierňujú efekt zmeny polohy zdvihu na hranici blokov, avšak efektívnejší spôsob spočíva v rozdelení roviny na „soft“ zóny, ktoré sa riadi zásadou priestorového filtrovania a odberu vzoriek s nízkym priechodom. Funkcia rozmazania podľa Iijima [9] predpokladá priestorové filtrovanie bez odberu vzoriek. Pri implementácii rozmazania sa funkcia impulznej odozvy (IRF) priestorového filtra aproximuje do váženého okna, ktoré sa tiež nazýva maska rozmazania. IRF je často gaussovská funkcia (6):

$$h(x,y) = \frac{1}{2\pi\sigma_x^2}\exp\left(-\frac{x^2+y^2}{2\sigma_x^2}\right). \tag{6}$$

Podľa teórie vzorkovania sa parameter rozptylu σ_x týka vzorkovacej frekvencie (recipročného intervalu vzorkovania). Pri skrátení šírky pásma gaussovského filtra bol uvedený empirický vzorec (7):

$$\sigma_x = \frac{\sqrt{2t_x}}{\pi} \tag{7}$$

kde t_x je interval vzorkovania. V mieste (x_0, y_0) obrázka $f(x, y)$ dáva konvolúcia vzorkovanú hodnotu funkcie (8):

$$F(x_0,y_0) = \sum_x \sum_y f(x,y)\cdot h(x-x_0, y-y_0). \tag{8}$$

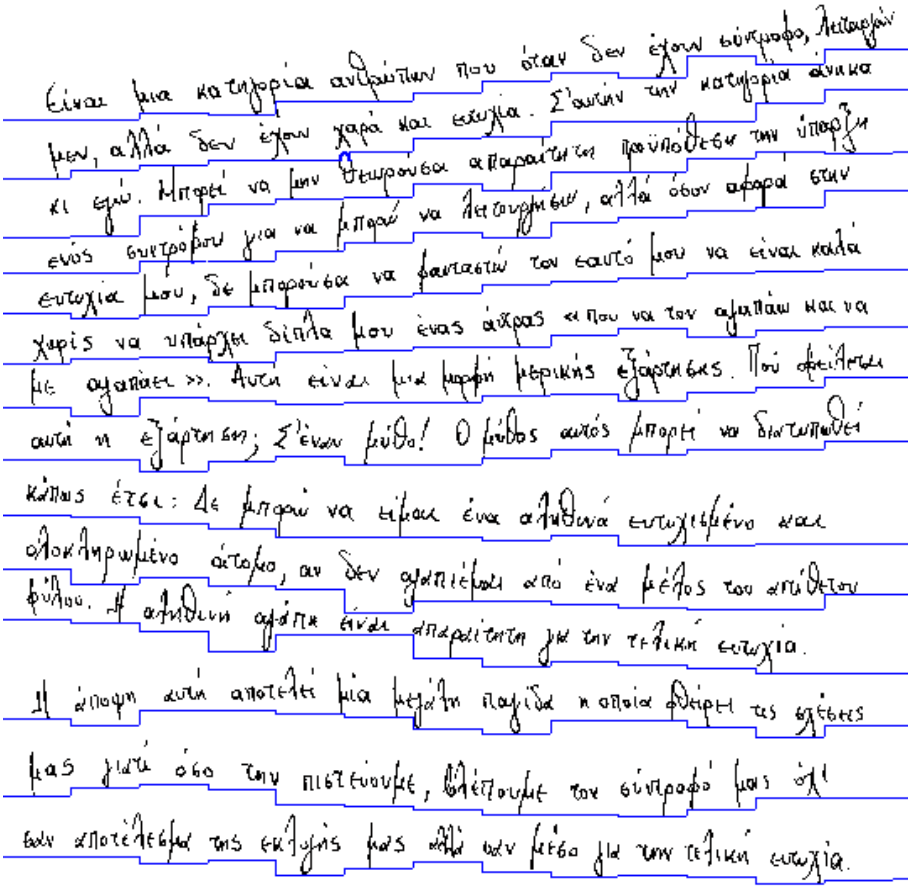
[10].

3 SEGMENTÁCIA TEXTU

Všetky segmentácie textových obrázkov je možné dosiahnuť na troch úrovniach. Keď sa pohybujeme na rôznych úrovniach hierarchie segmentácie textu, získame špecificky jemnejšie detaily. Používanie všetkých troch úrovní nie je povinné. Segmentácia na ktorejkoľvek z týchto úrovní priamo závisí od charakteru aplikácie [11].

3.1 Segmentácia riadkov

Segmentácia riadkov je prvý a primárny krok segmentácie textu. Zahŕňa horizontálne skenovanie obrázka, riadok po pixeli po rade zľava doprava a zhora nadol. Testuje sa intenzita každého pixelu. Na základe hodnôt pixelov zoskupujeme pixely do viacerých oblastí z celého obrázka. Rôzna oblasť označuje iný obsah v súbore obrázka. Následne je možné extrahovať požadovaný obsah [11]. Metódy pre segmentáciu riadkov sú napríklad: Script independent handwritten text line segmentation using Active contour, Language independent text line extraction using seam carving, alebo A Hough based algorithm for extracting text lines in handwritten documents [19].



Obrázok 12: Segmentácia riadkov [12]

3.2 Segmentácia slov

Segmentácia slov je proces analýzy zreťazeného textu, aby sa zistilo, kde existujú zlomky slov. Použitím vertikálneho projekčného profilu je možné získať súčet stĺpcov. Hľadaním miním v horizontálnom projekčnom profile jedného riadka môžeme oddeliť jednotlivé slová [12]. Metódy pre segmentáciu slov sú napríklad: Pivot-based search for word spotting in archive documents, alebo Local gradient histogram features for word spotting in uncostrained handwritten document, ktoré pozostáva z dvoch odlišných systémov pre rozpoznávanie slov – Hidden Markov Model a Dynamic time wrapping [19].

3.3 Segmentácia znakov

Považuje sa za jeden z hlavných krokov predspracovania v rukopisoch písaných kurzívou, ako sú arabské, urdské a ďalšie rukopisy, v ktorých sú znaky navzájom spojené. Preto existuje veľa techník [8], ako napríklad Multi oriented and Multi sized touching character segmentation using Dynamic Programming, Segmentation of Isolated and touching characters in offline Handwritten Gurumukhi Script recognition, Handwritten text segmentation using Average longest path algorithm, alebo Segmentation of off-line cursive Handwriting using Linear Programming [19], vyvinutých pre segmentáciu znakov a väčšina z nich je špecifická pre určitý rukopis a nemusí byť vhodná pre iné rukopisy. Aj v tlačených rukou písaných dokumentoch je segmentácia znakov potrebná kvôli dotyku znakov pri písaní rukou. Napríklad tlačené latinské znaky sa dajú ľahko segmentovať pomocou horizontálnych a vertikálnych profilov histogramu. Menšie písma a písma, ktoré obsahujú pätky, však môžu spôsobiť dotyk, ktorý bude potrebovať ďalšie spracovanie, aby sa vyriešil problém dotýkania slov [8].

Obrázok 13: Správna segmentácia slova "Buchanan" [58]

3.3.1 Explicitná segmentácia

Pri explicitnej segmentácii je obraz vstupného slova postupnosti znakov rozdelený na čiastkové obrazy jednotlivých znakov, ktoré sú potom klasifikované. Tento proces sa nazýva pitva. Prístup vertikálnej segmentácie spočíva v kategórii explicitnej segmentácie. Pri tomto

postupe sa po predbežnom spracovaní vstupného, rukou písaného slova obrázok slova skenuje zhora nadol. Pozície všetkých týchto stĺpcov, ktorých súčet čiernych pixelov v popredí je buď 0 alebo 1 sa uložia. Tieto stĺpce sa označujú ako PSC (Potential segmentation columns). V obrázkoch slov každý stĺpec, pre ktorý je súčet popredných pixelov 0 alebo 1 je PSC. Použitie PSC vertikálne oreže obraz slova, ako je znázornené na obrázku číslo 5. Všetky PSC, ktorých vzdialenosť je menšia ako prahová hodnota, sa integrujú do jedného stĺpca [13].

Algoritmus predbežného spracovania a algoritmus segmentácie znakov segmentuje znaky zo slov, aby vytvoril zlomený znak, ktorý možno rozdeliť na znaky. Pozostáva z nasledujúcich krokov:

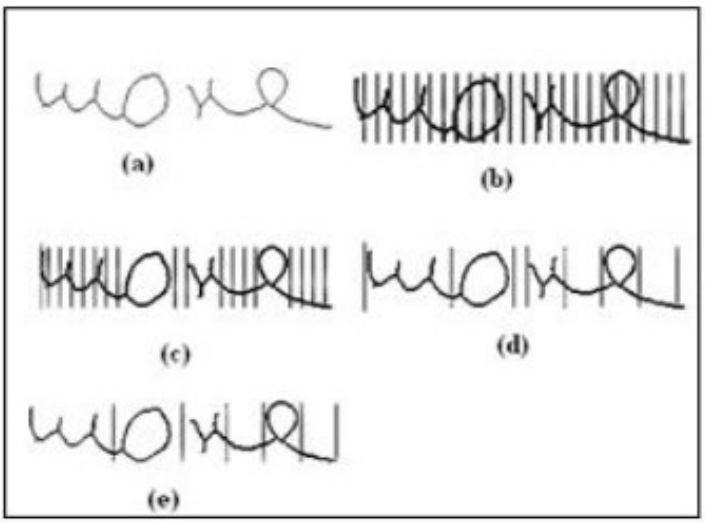
1. výber rôznych vzoriek kurzívneho rukopisného skriptu zo štandardnej databázy,
2. konverzie zhromažďovania dokumentov vo formáte PNG a ich konvertovanie do binárneho formátu,
3. konverzia binárnych obrázkov do 0 a 1, aby sa naň aplikovala detekcia okrajov,
4. po nájdení hrán sa vypracuje technika projekcie profilu histogramu,
5. po použití techniky projekcie profilu histogramu dostaneme výrezové body znaku vo vzdialenosti 20 pixelov [13].

Obrázok 14: Vzorky slov po segmentácii [13]

3.3.2 Implicitná segmentácia

Nazýva sa aj segmentácia založená na rozpoznávaní. V tejto metóde sa segmentácia a rozpoznávanie znakov dosahujú súčasne. Systém vyhľadá v obrázku súčasti, ktoré trieda obrázkov zobrazuje vo svojej abecede. Implicitným segmentačným postupom je rozdeliť slová na segmenty, ktoré by mali byť znakmi, a potom každý segment odovzdať klasifikátoru [13].

Implicitná segmentačná metóda poskytuje všetky predbežné segmenty a umožňuje klasifikátoru rozhodnúť sa o najlepšej hypotéze segmentácie. Pri výbere počtu segmentov pre slovo však existuje kompromis. Menší počet segmentov je základom efektívneho výpočtu, zatiaľ čo veľký počet segmentov je výpočtovo náročný. Implicitná segmentačná metóda je dvojstupňový proces. V prvej fáze sa na segmentáciu slov použije schéma segmentácie založená na heuristických pravidlách. V druhej fáze sa vykonáva overovanie [13].



Obrázok 15: Implicitná segmentácia [13]

3.3.3 Holistická segmentácia

Holistická metóda segmentácie je známa ako metóda bez segmentácie [13]. Holistický proces rozpoznáva celé slovo ako jednotku. Hlavnou nevýhodou tejto triedy metód je to, že ich použitie je zvyčajne obmedzené na vopred definovaný lexikón. Keďže sa nezaoberajú priamo písmenami, ale iba slovami, rozpoznávanie sa nevyhnutne obmedzuje na konkrétny lexikón slov. Tento bod je mimoriadne dôležitý, keď sa vyžaduje školenie o vzorkách slov. Fáza trénovania je preto povinná rozšíriť, alebo upraviť lexikón možných slov. Táto vlastnosť robí tento druh metódy vhodnejšou pre aplikácie, v ktorých je lexikón staticky

definovaný, napríklad pri rozpoznávaní bankových šekov. Môže sa tiež použiť na online rozpoznávanie na osobnom počítači, pričom rozpoznávací algoritmus sa potom vyladí na písanie konkrétneho používateľa, ako aj na príslušnú slovnú zásobu [14].

Tabuľka 1: Porovnanie rôznych metód segmentácie [19], kvôli pôvodnosti a dohľadateľnosti nie sú metódy preložené do slovenského jazyka

Dokument	**Metóda**	**Presnosť**	**Data set**
Language independent text line extraction using seam carving	Seam Carving	98,9%	Arabic, Chinese and English historical documents, ICD AR2007
Text line and word segmentation of handwritten documents	Hough transform, Euclidean distance and convex hull metric	Modern set – 97,4%, Historical data – 99,1%	Modern set- ICDAR2007, Hitorical set – 40 images taken from historical archives of university of Athens
Pivot-based Search for Word Spotting in Archive Documents	Pivot based search algorithm	78%	22 pages from a book of census of a medieval city
Local gradient histogram features for word spotting in unconstrained handwritten document	Hidden Markov models and Dynamic time warping	Better performance than state-of-art	630 real scanned letters (written in French) submitted to the customer department of a company
HMM based offline English character recognition	Multiple hidden markov model(HMM)	98,26%	Database of 13000 samples from 100 writers

4 EXTRAKCIA PRÍZNAKOV

Extrakcia príznakov je metóda, ktorá hodnotí jedinečnú identitu ľubovoľného objektu alebo vzoru. Vo fáze extrakcie príznakov je každý znak reprezentovaný ako vektor prvku, ktorý sa stáva jeho identitou. Hlavným cieľom extrahovania príznakov je extrahovať stanovené vlastnosti, ktoré maximalizujú mieru rozpoznávania s najmenším počtom prvkov [15].

Extrakcia prvkov je potrebná na zníženie množstva údajov do vektora prvkov. Napríklad akýkoľvek lekársky röntgenový obrázok vždy obsahuje milióny pixelov, ktoré sa musia extrahovať ako príznak. Cieľom procesu extrakcie príznakov je filtrácia informácií do koncentrovanejšej a zvládnuteľnejšej formy, ktorá je známa aj ako vektor príznaku obrazu. Nástroj, ktorý sa používa na extrakciu maximálneho príznaku abecedy, je potrebou každého klasifikátora na jeho efektívny tréning. V prípade ručne písaného charakteru je ťažké získať tieto príznaky z dôvodu divertívnej alebo kurzívnej povahy rukopisu, pretože obsahuje vysoký stupeň variability a nepresnosti [15].

Metódy extrahovania príznakov sú založené na troch typoch funkcií:

1. štatistické,
2. štrukturálne,
3. globálne transformácie a momenty [15].

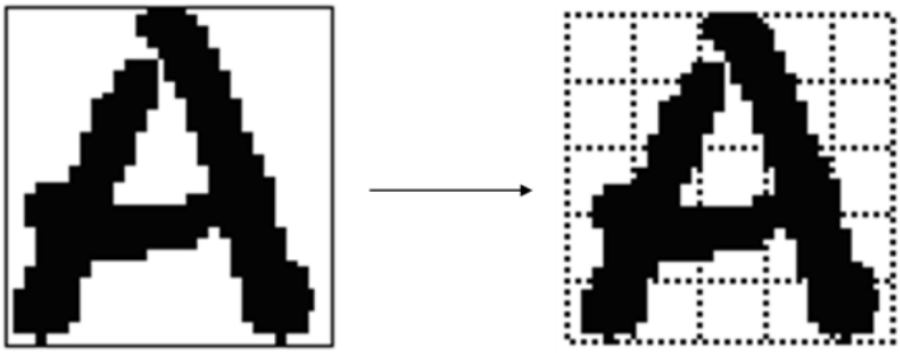
4.1 Štatistické príznaky

Reprezentácia izolovaného obrazu znaku štatistickými distribučnými bodmi by sa mala do istej miery starať o zmeny štýlov [15]. Niektoré z hlavných štatistických príznakov používaných na reprezentáciu znakov sú:

1. zoning,
2. projekcie a profily,
3. kríženia a vzdialenosti [16].

4.1.1 Zoning

Znakový obrázok je rozdelený do NxM zón. Z každej zóny sú príznaky extrahované do podoby vektora príznakov. Cieľom Zoningu je získať lokálne charakteristiky namiesto globálnych charakteristík [16].



Obrázok 16: Obrázok pred Zoningom a po Zoningu [17]

4.1.2 Projekčné histogramy

Základnou myšlienkou používania projekcií je to, že charakterové obrazy, ktoré sú 2-D signálmi, môžu byť reprezentované ako 1-D signál. Tieto vlastnosti, hoci sú nezávislé od hluku a deformácie, závisia od rotácie. Projekčné histogramy počítajú počet pixelov v každom stĺpci a riadku obrázka znaku. Projekčné histogramy môžu oddeliť znaky ako „m“ a „n“ [17].

Obrázok 17: Štatistická metóda:
Projekčný histogram [17]

4.1.3 Profily

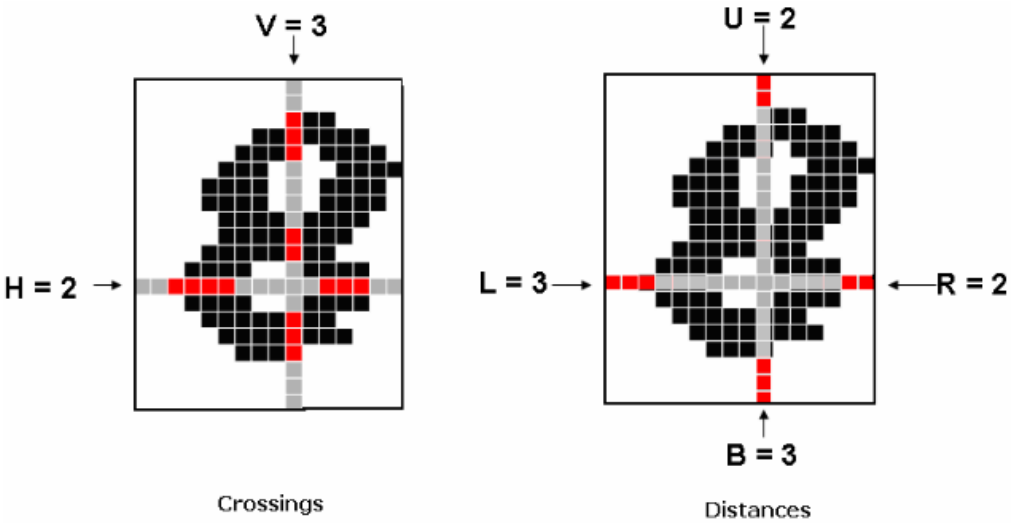
Profil počíta počet pixelov (vzdialenosť) medzi hraničným rámčekom obrázka znaku a okrajom znaku. Profily dobre opisujú vonkajšie tvary znakov a umožňujú rozlišovať medzi veľkým počtom písmen, napríklad „p“ a „q“ [17].

Obrázok 18: Štatistická metóda: Profily [17]

4.1.4 Kríženia a vzdialenosti

Kríženia počítajú počet prechodov z pixelov pozadia na pixely znaku – popredie. Kríženia sa počítajú po vertikálnych a horizontálnych riadkoch obrázka.

Vzdialenosti vypočítajú vzdialenosti prvého obrázkového pixelu detekovaného od horného a dolného okraja obrázka, pozdĺž vertikálnych čiar a od ľavého a pravého okraja pozdĺž horizontálnych čiar [17].

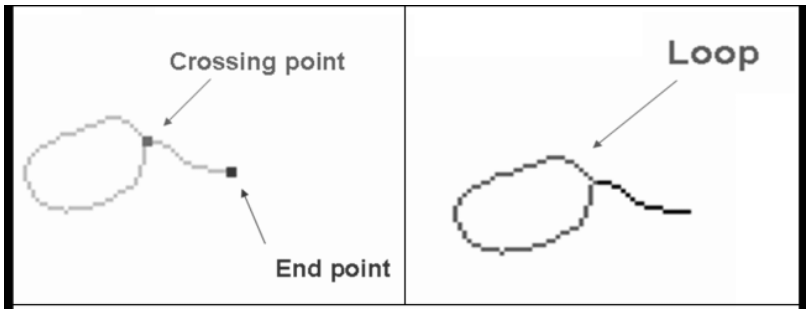


Obrázok 19: Kríženia a vzdialenosti [17]

4.2 Štrukturálne príznaky

Znaky môžu byť reprezentované štrukturálnymi príznakmi s vysokou toleranciou k deformáciám a variáciám štýlov [17]. Tento typ reprezentácie môže tiež zakódovať určité znalosti o štruktúre objektu alebo môže poskytnúť určité znalosti o tom, aké druhy

komponentov tvoria tento objekt. Štrukturálne príznaky sú založené na topologických a geometrických vlastnostiach znaku, ako sú napríklad pomer strán, priečne body, slučky, vetvené body, ťahy a ich smery, inflexia medzi dvoma bodmi, vodorovné krivky vrchné alebo spodné atď. [18].



Obrázok 20: Štrukturálna metóda [17]

4.3 Globálne transformácie – momenty

Vypočíta sa Fourierova transformácia obrysu obrázka. Pretože prvé *n* koeficienty transformácie sa môžu použiť na rekonštrukciu obrysu, tieto *n* koeficienty sa považujú za *n*-rozmerný príznakový vektor, ktorý predstavuje znak. Vďaka Centrálnym a Zenrike momentom, je proces rozpoznávania mierky objektu, prekladu a rotácie nemenný. Pôvodný obrázok je možné kompletne zrekonštruovať z momentových koeficientov [18].

5 POST-PROCESSING

Post-processing je záverečná fáza, ktorá zlepšuje rozpoznávanie tým, že upresňuje rozhodnutia prijaté v predchádzajúcej etape a rozpoznáva slová pomocou kontextu. V konečnom dôsledku je post-processing zodpovedný za výstup najlepšieho riešenia a je často implementovaný ako súbor techník, ktoré sa spoliehajú na frekvenciu znakov, slovníky a ďalšie kontextové informácie [22].

Keďže klasifikácia niekedy vytvára možné riešenia namiesto jedinečného riešenia, následné spracovanie je zodpovedné za výber správneho riešenia pomocou informácií na vyššej úrovni, ktoré klasifikátor nemá k dispozícii. Post-processing tiež používa tieto informácie na vyššej úrovni na kontrolu správnosti riešení vrátených klasifikátorom. Najbežnejšie operácie po spracovaní sú kontrola a korekcia pravopisu. Kontrola pravopisu môže byť rovnako jednoduchá ako vyhľadávanie slov v slovníku [22].

6 DATASETY

Dataset je súbor dát. Inými slovami, množina údajov, ktorá zodpovedá obsahu jednej databázovej tabuľky alebo jednej matice štatistických údajov, pričom každý stĺpec tabuľky predstavuje konkrétnu premennú a každý riadok zodpovedá danému členovi príslušnej množiny údajov [60].

V projektoch strojového učenia sú potrebné tréningové sady, ktoré sa používajú na trénovanie modelu a testovacie sady [60].

Tréningová sada sa používa na trénovanie algoritmu, ako napríklad neurónové sieť, aby sa naučila ako klasifikovať údaje z testovacej sady. V prípade učenia s učiteľom (supervised) obsahuje vstupné dáta a očakávaný výstup [60].

Testovacia sada sa používa na vyhodnotenie toho, ako dobre bol algoritmus naučený zo súboru tréningovej sady [60].

Tabuľka 2 zobrazuje súhrnný prehľad presností, pre klasifikátory popísané v kapitole 1, získaných z vedeckých publikácií a zdrojov, podrobnejšie definovaných ďalej.

Tabuľka 2: Porovnanie klasifikátorov na rôznych datasetoch

Datasety	**NB**	**k-NN**	**SVM**	**MLP**	**LSTM**	**CNN**
MNIST	81,54%	97,33%	97,74%	98,27%	99,22%	99,48%
EMNIST	X	88,56%	89,51%	87,79%	X	93,63%
Arabské čislice	81,10%	92,90%	95,20%	96,00%	X	97,40%
Arabské znaky	X	95,30%	96,30%	X	X	97,70%
Oriya číslice	85,50%	91,50%	94,20%	95,20%	X	X
Bengali číslice	86,30	90,80%	94,30%	95,10%	X	98,45%
Bangla číslice	X	X	95,05%	98,20%	98,25%	97,93%
Bangla znaky	X	71,58%	80,51%	80,58%	87,41%	91,81%
Devanagari znaky	X	97,05%	98,35%	X	X	96,09%
Gurmukhi znaky	X	94,12%	94,29%	98,96%	X	99,34%

6.1 MNIST

Databáza MNIST bola vytvorená zo špeciálnej databázy NIST Special Database 3 a Special Database 1, ktoré obsahujú binárne obrázky rukou písaných číslic. NIST pôvodne navrhla SD-3 ako ich tréningovú sadu a sadu SD-1 ako testovaciu. Sada SD-3 je však omnoho čistejšia a ľahšie rozpoznateľná ako SD-1. Dôvodom je skutočnosť, že SD-3 sa zbierala medzi zamestnancami agentúry Census Bureau, zatiaľ čo SD-1 bola zozbieraná medzi študentami stredných škôl. Vyvodenie citlivých záverov z experimentov učenia si vyžaduje, aby bol výsledok nezávislý od výberu tréningovej a testu medzi kompletnou sadou vzoriek. Preto bolo potrebné vytvoriť novú databázu zmiešaním databáze NIST [23].

Trénovacia sada MNIST sa skladá z 30 000 vzorov zo sady SD-3 a 30 000 vzorov zo sady SD-1. 60 000 vzorových tréningových setov obsahuje príklady približne od 250 autorov [23].

Pre tento dataset dosiahla najlepší výsledok metóda CNN [37] s presnosťou 99,48%. Metóda LSTM [38] dosiahla presnosť 99,22% . Tretiu najvyššiu presnosť dosiahla metóda MLP [39], a to 98,27%. Presnosť 97,74% dosiahla metóda SVM [40], k-NN [40] dosiahla presnosť 97,33% a NB [41] 81,54%.

Obrázok 21: 100 ukážkových obrázkov
číslic z datasetu MNIST [24]

6.2 EMNIST

Dataset EMNIST je sada ručne písaných znakov číslic odvodených z databázy NIST Special Database 19 a skonvertovaných do obrazového formátu 28 x 28 pixelov a datasetovej štruktúry, ktorá priamo zodpovadá datasetu MNIST. Ďalšie informácie o obsahu datasetu a procese prevodu sú dostupné v dokumente [26], [25].

Najvyššiu presnosť 93,63% pre dataset EMNIST dosiahla metóda CNN [37]. Metóda SVM [40] dosiahla druhú najvyššiu presnosť 89,51%. Presnosť 88,56% bola dosiahnutá metódou k-NN [40] a 87,79% metódou MLP [37].

Obrázok 22: Vzorky všetkých písmen a číslic v súbore údajov EMNIST [37]

6.3 Arabské číslice

Dataset MADbase pozostáva z 70 000 obrázkov arabských číslic napísaných 700 účastníkmi z rôznych oblastí práce a zázemia. Sú rozdelené do tréningovej sady, ktorá obsahuje 60 000 obrázkov a testovacej sady, ktorá obsahuje 10 000 obrázkov [27]. Každý účastník napísal každú číslicu (od 0 do 9) desaťkrát. Aby sa zabezpečilo zahrnutie rôznych štýlov písania, databáza bola zozbieraná z rôznych inštitúcií: inžinierskych a právnych vysokých škôl, lekárskej fakulty, Open University (ktorej študenti majú široký vekový rozsah), vysokej školy a vládnej inštitúcie [28].

Pri porovnávaní arabských číslic dosiahla najväčšiu presnosť metóda CNN [43] s úspešnosťou 97,40%. Metóda MLP [42] dosiahla presnosť 96,00%. Tretiu najvyššiu presnosť 95,20% dosiahla metóda SVM [42]. Presnosť 92,90% dosiahla metóda k-NN [42] a najnižšiu presnosť dosiahla metóda NB [42] 81,10%.

6.4 Arabské znaky

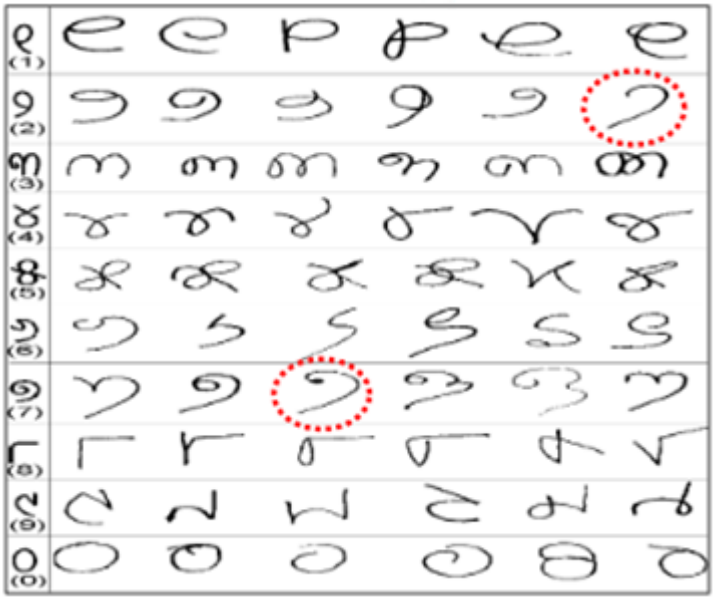
Dataset sa skladá zo 16 800 znakov napísaných 60 účastníkmi, vekového rozsahu od 19 do 40 rokov a 90% účastníkov písalo pravou rukou. Každý účastník napísal každý znak (od „alef“ po „yeh“) desaťkrát do dvoch formulárov. Formuláre boli skenované v rozlíšení 300 dpi. Každý blok je automaticky segmentovaný pomocou Matlab 2016a na určenie súradníc pre každý blok. Databáza je rozdelená na dve sady. Tréningová sada obsahuje 13 440 znakov a testovacia sada obsahuje 3 360 znakov. Autori tréningovej a testovacej sady sú divergentní. Pre zabezpečenie variability testovacej sady je usporiadanie zaradenia autorov do testovacej sady náhodné, aby sa zabezpečilo, že autori testovacej sady nepochádzajú z jednej inštitúcie [29].

Pre tento dataset dosiahla najlepší výsledok 97,7% metóda CNN [44]. Druhú najvyššiu presnosť dosiahla metóda SVM [44] 95,20% a najnižšiu presnosť 95,30% dosiahla metóda k-NN [44].

6.5 Oriya číslice

Oriya je populárny jazyk a písmo vo východnej časti Indie. Rovnako ako ostatné písma má aj Oriya 10 číslic. Pre získanie predstavy o Orijských číslach a ich variabilnosti v písaní rukou, je na obrázku 23 znázornená jedna sada tlačených a šesť sád ručne písaných číslic. Štrukturálny tvar Oriya čísiel sa líši od arabských a indických písiem. Hlavnou ťažkosťou pri rozpoznávaní Oria čísiel je variabilita ich tvaru. Z obrázku je zrejmé, že číslica môže mať dve, alebo viac tvarových variácií a jedna číslica sa môže vzhľadom na ich tvarové variácie javiť veľmi blízko inej číslici. Napríklad posledná číslica dva a tretia číslica sedem označené červeným krúžkom na obrázku 23. Tvar týchto dvoch číslic je veľmi podobný a táto tvarová podobnosť robí rozpoznávací systém komplikovanejším [30].

Najvyššiu presnosť pre tento dataset dosiahla metóda MLP [42] s úspešnosťou 95,20%. Úspešnosť 94,20% dosiahla metóda SVM [42]. Metóda k-NN [42] dosiahla presnosť 91,50% a metóda NB [42] dosiahla 85,50%.



Obrázok 23: Príklady Oriya ručne písaných číslic [30]

6.6 Bengali Handwritten Digits Database

Dataset BHDDB bol zozbieraný od študentov z Department of Computer Science and Engineering of Bangladesh University of Engineering and Technology. Študenti dostali formulár na zapísanie číslic. Formuláre mali pravidelnú mriežku, v ktorej boli číslice napísané. Formuláre sa skenovali farebne a mali vo svojich štyroch rohoch značku, ktorá sa použila na zarovnanie okrajov obrázka s čiarami mriežky [31].

Pre dataset ručne písaných Bengali číslic dosiahla najvyššiu presnosť metóda CNN [45] 98,45%. Metóda MLP [42] dosiahla presnosť 95,10%. Presnosť 94,30% dosiahla metóda SVM [42]. Štvrtú najvyššiu presnosť 90,80% dosiahla metóda k-NN [42] a metóda NB [42] dosiahla 86,30%.

6.7 Bangla číslice

Napriek tomu, že je jazyk Bangla bohatým jazykom, je datasetov ručne písaných Bangla čísiel veľmi málo. Indian Statistical Institut, Kolkata a Jadavpur University, Kolkata vypracovali dva nezávislé číselné datasety. Okrem týchto datasetov bol vyvinutý ďalší kompletný izolovaný dataset [32]. Okrem toho sú Banglalekha a CMATER databáza open-source datasety a obsahujú 6 000 a 19 748 izolovaných vzoriek čísiel. Na druhej strane je súbor údajov ISI väčší ako ostatné existujúce číselné datasety a obsahuje približne 23 299 izolovaných vzoriek rozdelených do tréningových a testovacích sád [33].

Pre tento dataset dosiahla najvyššiu presnosť metóda LSTM [46] s úspešnosťou 98,25%. Presnosť 98,20% dosiahla metóda MLP [46], 97,93% metóda CNN [46] a 95,05% metóda k-NN [46].

6.8 Bangla znaky

Znaková sada je rozdelená do dvoch kategórií: základné a zložené znaky. Základnými znakmi sú zbierky samohlások a spoluhlások. Bangla má 11 samohlások a 39 spoluhlások. Väčšina samohlások má jeden alebo viac kaligrafických tvarov, ktoré môžu byť pripojené k spoluhláskam v rôznych pozíciách. V prípade, že je samohláska pridaná k spoluhláske, tvar samohlásky sa zmení a zmenený znak sa nazýva modifikovaný znak, čo je vidieť na obrázku číslo 24 [35].

Najvyššiu presnosť 91,81% dosiahla metóda CNN [48]. Metóda LSTM [48] dosiahla druhú najvyššiu presnosť 87,41%. Metóda MLP [47] dosiahla presnosť 80,58%, SVM [63] 80,51% a k-NN [49] metóda dosiahla presnosť 71,58%.

Vowel	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ
Modifier	া	ি	ী	ু	ূ	ৃ	ে	ৈ	ো	ৌ
When attached to ক	কা	কি	কী	কু	কূ	কৃ	কে	কৈ	কো	কৌ

Obrázok 24: Modifikované znaky samohlásky [35]

6.9 Devanagari znaky

Bolo vygenerovaných 92 000 obrázkov pre Devanagari Handwritten Character Dataset, zobrazením znakov napísaných mnohými jednotlivcami, čo malo za následok veľké rozdiely v spôsobe, akým je každý znak napísaný. Autori naskenovali stovky ručne napísaných dokumentov od rôznych autorov a každý znak orezali ručne. Každý obrázok v datasete je jedinečný. Dataset je náhodne rozdelený do tréningovej sady (85%) a testovacej sady (15%). Tréningová sada sa skladá zo 78 200 obrázkov a testovacia sada pozostáva zo zostávajúcich 13 800 obrázkov. Každý obrázok má veľkosť 32 x 32 pixelov a znak je vycentrovaný do veľkosti 28 x 28 pixelov. Na všetkých štyroch stranách obrázka bolo pridané vnútorné odsadenie dvoch pixelov s hodnotou 0 [34].

Pre tento dataset dosiahla najvyššiu presnosť 98,35% metóda SVM [50]. Druhú najvyššiu presnosť 97,05% dosiahla metóda k-NN [50] a metóda CNN [50] dosiahla presnosť 96,09%.

6.10 Gurmukni znaky

Jazyk Gurmukhi je jazyk používaný na písanie pandžábskeho jazyka a je odvodený od starého pandžábskeho výrazu „Guramukhi“, čo znamená „z úst gurua“. Gurmukhi jazyk pozostáva z troch samohlások, tridsiatich dvoch spoluhlások, šiestich dodatočných spoluhlások, deviatich modifikátorov samohlások, troch pomocných znakov a troch polovičných znakov. Je dvanásty najrozšírenejší jazyk na svete. Štýl písania je zhora nadol a zľava doprava. V jazyku Gurmukhi nie je citlivosť na veľké a malé písmená [36].

Pri rozoznávaní znakov jazyka Gurmukni dosiahla najlepší výsledok 99,34% metóda CNN [54]. Metóda MLP [52] dosiahla druhý najlepší výsledok 98,96%. Presnosť 94,29% získala metóda SVM [53] a 94,12% metóda k-NN [51].

II. PRAKTICKÁ ČASŤ

7 APLIKÁCIE PRE ROZPOZNÁVANIE PÍSANÉHO PÍSMA

Na GitHube[1] sa nachádza množstvo repozitárov s aplikáciami pre rozpoznávanie ručne písaného písma. Veľa z týchto repozitárov sú však len rozpracované projekty. Takisto sa tam nachádzajú projekty, ktoré používajú staršie verzie knižníc. Tieto knižnice obsahujú funkcie, ktoré nie sú zahrnuté v novších verziách, takže aby bola aplikácia spustiteľná, musela by sa prerobiť veľká časť kódu.

Aplikácia sú napísané v jazyku Python. Aplikácie nerónových sietí sú napísané v prostredí Jupyter Notebook. V aplikáciách neurónových sietí sa používajú epochy a batche. Jedna epocha znamená, keď celý dataset prejde obomi smermi neurónovej siete práve raz. Batch_size je počet tréningových vzoriek prezentovaných v jednom batch-i. Keďže nie je možné poslať celý dataset do neurónovej siete naraz, musí byť rozdelený na tzv. batche.

Funkčné aplikácie, boli spustené, otestované a sú popísané v nasledujúcich podkapitolách.

Aplikácie boli spustené na zariadení s:

- Procesor: Intel Core i7-7700HQ,
- RAM: 16gb,
- Grafická karta: NVIDIA GeForce GTX 1050 Ti.

Požiadavky pre spustenie aplikácií:

- Python 3.5+,
- Jupyter Notebook
- Skicit-Learn (najnovšia verzia),
- Numpy (+ mkl pre Windows),
- Matplotlib,
- Keras,
- TensorFlow,
- Pandas.

[1] Webová služba podporujúca vývoj softwaru za pomoci verzovacieho nástroja Git

V tabuľke 3 sa nachádza stručný prehľad použitých a testovaných aplikácií autorom bakalárskej práce a dosiahnuté presnosti klasifikátorov. V kolónke autor je uvedený autor kódu na GitHube. V ďalších podkapitolách sú uvedené detaily k jednotlivým aplikáciám.

Tabuľka 3: Výsledky testovaných aplikácií

Názov aplikácie	**Klasifikátor**	**Dataset**	**Presnosť**	**Autor**
Handwritten Digit Recognition using Deep Learning	k-NN	MNIST	96,80%	Anuj Dutt, Banee Ishaque K [65]
Handwritten Digit Recognition using Deep Learning	SVM	MNIST	97,86%	Anuj Dutt, Banee Ishaque K [65]
MLP-for-MNIST-dataset	MLP	MNIST	98,19%.	Amilineni Roshni Naidu [67]
RNN and LSTM for Image Classifications	LSTM	MNIST	99,07%	Saad Khattak [66]
Mnist_classification	CNN	MNIST	99,33%.	Ashwin Ram Kumar [64]

7.1 Handwritten Digit Recognition using Deep Learning

Aplikácia [65] pracuje s datasetom MNIST, ktorý je pre klasifikátor k-NN a SVM potrebné stiahnuť z webovej stránky [23]. Táto aplikácia generuje tzv. konfúznu maticu (Confusion Matrix), ktorá ukazuje úspešné a neúspešné zaradenie každej číslice. Oba algoritmy boli spustene 5 krát a výsledky boli spriemerované.

7.1.1 k-NN

Klasifikátor k-NN dosiahol strednú (mediánovú) hodnotu úspešnosti na datasete MINIST 96,80%. Tabuľka 4 znázorňuje konfúznu maticu z prvého spustenia aplikácie, kedy klasifikátor dosiahol presnosť 96,90%. V tabuľke je vidieť, že najčastejšie boli zamenené číslice 1 a 9. Číslica 1 bola zamenená za číslicu 7 v 21 prípadoch a 22 krát bola zamenená číslica 9 za číslicu 4.

Tabuľka 4: Konfúzna matica pre k-NN MNIST

Číslo	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	972	1	1	0	0	2	3	1	0	0
1	0	1133	2	0	0	0	0	0	0	0
2	11	7	991	2	0	0	2	15	4	0
3	0	3	3	977	1	12	1	6	4	3
4	2	8	0	0	942	0	4	3	1	22
5	6	0	0	10	1	863	3	2	2	5
6	5	3	0	0	3	2	945	0	0	0
7	0	21	3	0	2	0	0	992	0	10
8	7	4	8	13	6	11	4	6	911	4
9	5	6	3	7	7	4	1	10	2	965

7.1.2 Support Vector Machines

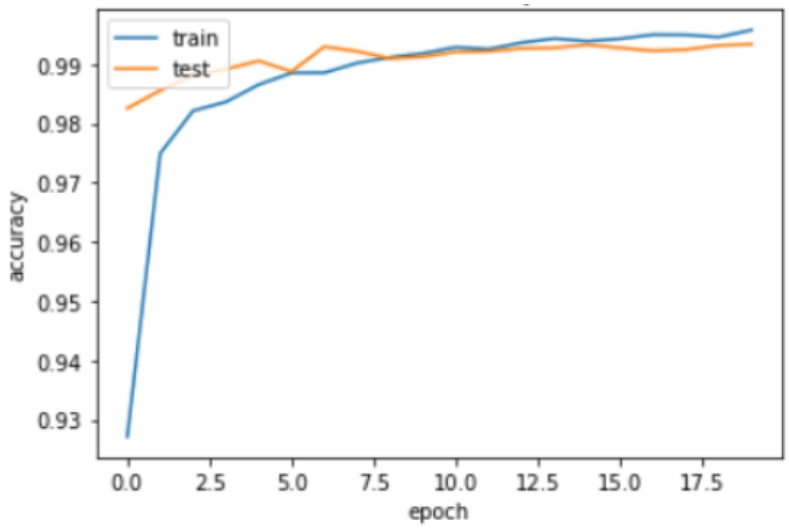
SVM klasifikátor dosiahol strednú hodnotu presnosť 97,86%. Tabuľka 5 zobrazuje konfúznu maticu z druhého spustenia aplikácie, kedy klasifikátor dosiahol najvyššiu úspešnosť 97,90%. Najčastejšie zamieňanou číslicou bola číslica 3 za číslicu 5 a to v 13 prípadoch. Druhými najčastejšie zamieňanými bola číslica 1 a 7, a to v 10 prípadoch. Číslica 2 bola zamieňaná za číslicu 7 a číslica 4 za číslicu 9.

Tabuľka 5: Konfúzna matica pre SVM MNIST

Číslo	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	972	0	1	0	0	3	1	1	2	0
1	0	1127	2	1	0	0	3	0	0	0
2	7	1	1007	0	2	0	5	7	3	0
3	0	2	1	990	0	5	0	4	5	3
4	2	0	1	0	968	0	3	0	0	8
5	2	0	1	13	1	867	4	1	2	1
6	5	5	1	0	3	6	936	0	2	0
7	0	10	9	2	1	0	0	1000	0	6
8	4	1	1	4	3	4	0	5	949	3
9	2	7	1	5	8	3	1	4	4	974

7.2 MNIST_classification

Autor tejto aplikácie [64] požíva CNN klasifikátor pre MNIST dataset importovaný z open-source knižnice TensorFlow. Dataset obsahuje 60 000 tréningových obrázkov a 10 000 testovacích obrázkov, ktoré majú rozmer 28 x 28 pixelov. Klasifikátor sa skladá z ôsmich blokov, ktoré pozostávajú z dvoch Conv2D vrstiev, dvoch Dense vrstiev, dvoch Dropout vrstiev, jednej Flaten vrstvy a jednej maxPooling2D vrstvy. Pôvodne používa 10 epoch. Pri tomto množstve dosiahol klasifikátor úspešnosť 99,16%. Po zvýšení počtu epoch na 20 dosiahol úspešnosť 99,33%. Pôvodný kód neobsahoval spracovanie výsledkov do konfúznej matice, preto bolo potrebné tento kód upraviť. Obrázok 24 ukazuje graf presnosti klasifikátora CNN.



Obrázok 25: Presnosť klasifikátora CNN

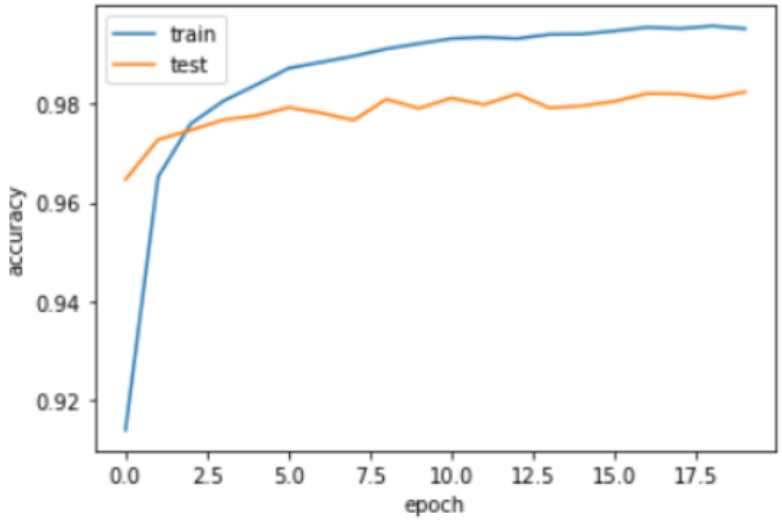
V tabuľke 6 je možné vidieť, že najčastejšie zamieňanými číslicami boli číslice 1, 3, 7 a 9. Číslica 3 bola zamieňaná za číslicu 5, číslica 1 a 7 za číslicu 2 a číslica 9 za číslicu 8.

Tabuľka 6: Konfúzna matica pre CNN MNIST

Číslo	0	1	2	3	4	5	6	7	8	9
0	977	0	0	0	0	0	0	0	2	1
1	0	1129	4	0	0	0	2	0	0	0
2	1	0	1027	0	0	0	1	3	0	0
3	0	0	1	1005	0	2	0	1	1	0
4	0	0	0	0	975	0	1	0	3	3
5	0	0	0	6	0	884	1	0	1	0
6	2	1	0	0	1	1	952	0	1	0
7	0	3	4	0	0	0	0	1019	1	1
8	2	0	1	1	0	0	0	1	969	1
9	1	0	0	1	3	0	0	3	4	996

7.3 RNN and LSTM for Image Classifications

Aplikácia [66] používa dataset MNIST, importovaný pomocou open-source knižnice TensorFlow. Klasifikátor je zložený z piatich blokov, ktoré obsahujú dve LSTM vrstvy, dve Dropout vrstvy a tri Dense vrstvy. Pôvodne sa v aplikácii používa 10 epoch a pri tomto množstve dosiahol klasifikátor LSTM presnosť 98,33% . Po zvýšení počtu epoch na 20, dosiahol klasfikátor úspešnosť 99,07%. Obrázok 26 predstavuje graf presnosti LSTM.



Obrázok 26: Presnosť klasifikátora LSTM

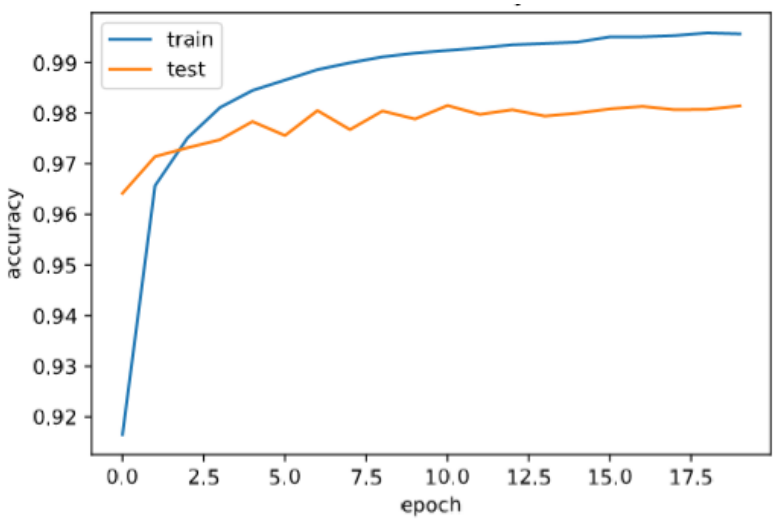
V tabuľke 7 je vidieť, že najčastejšie zamieňanou číslicou je číslica 9 za číslicu 4. Druhou najviac zamieňanou číslicou je číslica 3 za číslicu 2.

Tabuľka 7: Konfúzna matica pre LSTM MNIST

Číslo	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	977	0	0	0	0	0	1	1	1	0
1	0	1129	1	0	0	1	1	2	1	0
2	0	0	1027	0	1	0	0	3	0	0
3	0	0	6	1001	0	1	0	0	1	1
4	0	0	0	0	974	0	1	0	2	5
5	0	0	1	5	0	877	5	0	1	3
6	2	1	0	0	1	0	952	0	2	0
7	0	2	5	0	0	0	0	1017	0	4
8	0	0	1	0	0	1	1	0	970	1
9	1	0	0	0	15	0	0	5	5	983

7.4 MLP-for-MNIST-dataset

Aplikácia [67] používa MLP klasifikátor a dataset bol stiahnutý pomocou knižnice Keras a taktiež obsahuje obsahuje 60 000 tréningových obrázkov a 10 000 testovacích obrázkov. Klasifikátor obsahuje MaxPool2D vrstvu, Flatten vrstvu a tri Dense vrstvy. Pri pôvodnom počte 10 epoch dosiahol klasifikátor presnosť 97,93%. Po upravení počtu epoch na 20 sa podarilo zvýšiť presnosť na 98,19%. Na obrázku 27 je zobrazený graf úspešnosti klasifikátora.



Obrázok 27: Úspešnosť klasifikátora MLP

V tabuľke číslo 8 je vidieť, že najčastejšie zamieňanými číslicami sú číslice 4 a 9. Druhými najčastejšie zamieňanými sú číslice 7 a 2.

Tabuľka 8: Konfúzna matica pre MLP MNIST

Číslo	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	973	1	0	1	0	0	1	1	2	1
1	0	1130	2	0	0	1	1	0	1	0
2	4	0	1016	2	1	0	1	4	3	1
3	0	0	0	995	0	1	0	5	2	7
4	3	1	2	0	958	0	4	2	1	11
5	3	0	0	14	1	860	1	1	7	5
6	5	2	1	1	4	2	939	0	4	0
7	1	4	8	1	1	0	0	1004	2	7
8	2	0	3	5	2	2	1	3	952	4
9	2	3	0	2	5	0	0	2	2	992

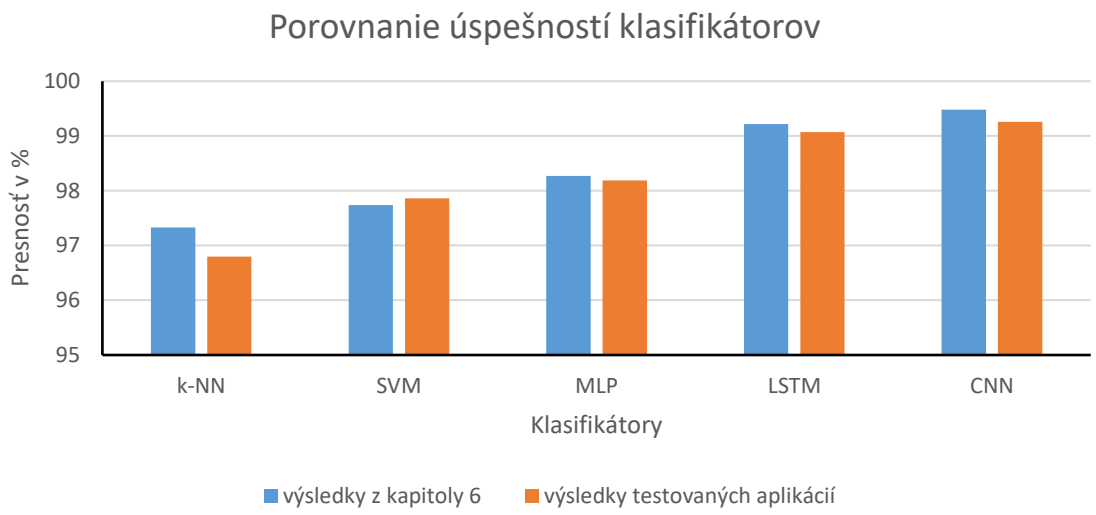
7.5 Zhodnotenie dosiahnutých výsledkov

Každá aplikácia bola odskúšaná na datasete MNIST, ktorý obsahoval 60 000 tréningových vzoriek a 10 000 testovacích vzoriek. Každý obrázok bol v rozlíšení 28 x 28 pixelov. Pre aplikácie s klasifikátormi k-NN a SVM bolo potrebné dataset stiahnuť z webovej stránky [23]. Ostatné aplikácie sťahovali datasety pomocou knižníc TensorFlow, alebo Keras. Nie všetky aplikácie vytvárali konfúzne matice, takže niektoré aplikácie bolo potrebné upraviť. Časovo menej náročné aplikácie na neučenie, konkrétne k-NN a SVM, boli spustené 5 krát a výsledky ich dosiahnutej presnosti boli spriemerované. Klasifikátory LSTM, MLP a CNN boli časovo náročnejšie kvôli trénovaniu neurónovej siete. Každý z klasifikátorov neurónovej siete mal autormi nastavený počet epoch na 10. Po zvýšení počtu epoch na 20 sa zvýšila presnosť každého klasifikátora. Klasifikátor CNN dosiahol presnosť vyššiu o 0,17%, klasifikátor MLP zvýšil svoju presnosť o 0,26% a klasifikátor LSTM pridaním epoch zvýšil presnosť až o 0,74%.

V porovnaní s úspešnosťou klasifikátorou v kapitole 6 dosiahli testované klasifikátory uspokojujúce výsledky. Rozdiel metód bol len v desatinách percenta. Metóda k-NN opísaná v kapitole 6 dosiahla presnosť 97,33%. Testovaná metóda dosiahla z piatich behov programu minimálnu presnosť 96,75%, maximálnu presnosť 96,90% a strednú hodnotu presnosti 96,80%. Testovaný SVM klasifikátor dosiahol dokonca vyššiu presnosť ako podľa

dokumentu [40]. Minimálna presnosť z piatich behov programu testovaného SVM bola 97,76%, maximálna presnosť bola 97,9% a stredná hodnota bola 97,86%. Presnosť z dokumentu bola 97,74%. MLP klasifikátor opísaný v dokumente [39] dosiahol 98,27% a testovaný dosiahol 98,19%. Druhým najpresnejším klasifikátorom podľa kapitoly 6 a aj podľa dosiahnutých výsledkov pri testovaní aplikácie je LSTM. V dokumente [38] je uvedená presnosť 99,22% a pri testovaní bola dosiahnutá presnosť 99,07%. Z dosiahnutých výsledkov pri testovaní klasifikátorov, ale aj podľa výsledkov v kapitole 6 je z popísaných klasifikátorov najvhodnejšia pre dataset MNIST metóda CNN. Podľa dokumentu [37] dosiahla táto metóda 99,48% a testovaná metóda dosiahla 99,33%.

Následujúci graf zobrazuje porovnanie úspešností klasifikátorov z kapitoly 6 s klasifikátormi z testovaných aplikácií.



Obrázok 28: Graf porovnaní úspešností klasifikátorov

Aj keď sa metóda CNN ukázala ako najúspešnejšia, jej trénovanie je časovo náročné. V prípade, že je potreba rýchlej klasifikácie, kedy sa nekladie dôraz na čo najpresnejší výsledok, je vhodné použiť aj klasifikátory k-NN a SVM.

ZÁVER

Cieľom mojej bakalárskej práce bolo vykonanie rešerše existujúcich metód pre rozpoznávanie rukou písaného písma a uskutočnenie prieskumu dostupných aplikácií pre rozpoznávanie a ich následné otestovanie. Zároveň táto práca môže slúžiť aj ako predpríprava k budúcej diplomovej práci, ktorá by sa zaoberala tvorbou aplikácie pre rozpoznávanie písma kurent, ktoré sa používalo od 16. storočia do polovice 20. storočia napríklad v matrikách.

V prvej kapitole teoretickej časti popisujem klasifikačné algoritmy, ktoré sa používajú na mapovanie vstupných údajov do konkrétnej triedy, alebo kategórie. Ďalej sú popísané metódy predspracovania obrazu, ktorých hlavným účelom je odstrániť z obrazu informácie, ktoré môžu negatívne ovplyvniť rozpoznávanie, ako napríklad šum, alebo zošikmenie. Ďalšie kapitoly popisujú segmentáciu textu, metódy extrakcie príznakov a post-processing. Segmentácia, alebo rozdelenie textu pozostáva zo segmentácie riadkov, slov a nakoniec písmen. Cieľom extrakcie príznakov je extrahovanie vlastností vstupného obrázka, pre maximalizovanie miery rozpoznávania. V poslednej kapitole teoretickej časti popisujem, s využitím odborných publikácií, konkrétne datasety a výsledky klasifikátorov z prvej kapitoly, na týchto datasetoch.

Praktická časť mojej práce sa venuje prieskumu a testovaniu dostupných aplikácií pre rozpoznávanie rukou písaného písma. Aj keď sa na GitHube nachádza množstvo aplikácií, nie všetky sa podarilo spustiť, kvôli starším verziám knižníc, ktoré používajú. Kapitola 7 je zameraná na aplikácie, ktoré sa podarilo spustiť. Každá aplikácia bola spustená a testovaná na datasete MNIST a vytvára tzv. konfúznu maticu, ktorá ukazuje úspešné a neúspešné zaradenie každej číslice. Dataset MNIST obsahuje 60 000 tréningových a 10 000 testovacích obrázkov. Všetky testované algoritmy dosiahli veľmi uspokojivé výsledky v porovnaní s výsledkami z kapitoly 6. Rozdiel vo výsledkoch testovaných algoritmov a algoritmov zo 6. kapitoly bol v desatinách percenta. Algoritmy neurónových sietí boli časovo náročnejšie na trénovanie, než algoritmy k-NN a SVM, avšak dosiahli lepšie výsledky. Najlepšie výsledky dosiahla metóda CNN. Pri autorom nastavených 10 epochách dosiahla táto metóda 99,16%. Po zvýšení počtu epoch na 20 dosiahla metóda 99,33%. Najrýchlejšou, ale najmenej úspešnou metódou z testovaných aplikácií je metóda k-NN, ktorá z piatich behov programu dosiahla maximálnu úspešnosť 96,90%. Z dosiahnutých výsledkov vyplýva, že ak je cieľom rozpoznávania čo najväčšia presnosť a nekladie sa dôraz na čas rozpoznávania, najvhodnejšou metódou je metóda konvolučných neurónových sietí. Na druhú stranu, ak sa

uprednostňuje rýchlosť pred presnosťou rozpoznávania, je vhodnejšie použiť algoritmy k-NN a SVM, nakoľko oba algoritmy dosiahli úspešnosť viac ako 96%.

ZOZNAM POUŽITEJ LITERATÚRY

[1] BEZZERA, Byron et al., 2017. *Handwriting: recognition, development and analysis*. 1. New York: Nova Science Publishers. ISBN 9781536119374.

[2] JAIN, Rashmi, 2017. Simple Tutorial on SVM and Parameter Tuning in Python and R. *Hackerearth* [online]. [cit. 2020-08-02]. Dostupné z: https://www.hackerearth.com/blog/developers/simple-tutorial-svm-parameter-tuning-python-r

[3] NASSAN, Mohamed, Abdelazim NEGM a Mohamed ZAHRAN, 2015. ASSESSMENT OF ARTIFICIAL NEURAL NETWORK FOR BATHYMETRY ESTIMATION USING HIGH RESOLUTION SATELLITE IMAGERY IN SHALLOW LAKES: CASE STUDY EL BURULLUS LAKE. *International Water Technology Journal* [online]. **5**(4), 248-259 [cit. 2020-08-02]. Dostupné z: https://www.researchgate.net/publication/303875065_ASSESSMENT_OF_ARTIFICIAL_NEURAL_NETWORK_FOR_BATHYMETRY_ESTIMATION_USING_HIGH_RESOLUTION_SATELLITE_IMAGERY_IN_SHALLOW_LAKES_CASE_STUDY_EL_BURULLUS_LAKE

[4] POLETAEV, I E, K S PERVUNIN a M P TOKAREV, 2016. Artificial neural network for bubbles pattern recognition on the images. *Journal of Physics: Conference Series* [online]. **754** [cit. 2020-08-02]. DOI: 10.1088/1742-6596/754/7/072002. ISSN 1742-6588. Dostupné z: https://iopscience.iop.org/article/10.1088/1742-6596/754/7/072002

[5] PONNUSAMY, Poovizhi, 2014. A STUDY ON PREPROCESSING TECHNIQUES FOR THE CHARACTER RECOGNITION. *International Journal of Open Information Technologies* [online]. **2**(12) [cit. 2020-08-02]. ISSN 23078162. Dostupné z: http://search.ebscohost.com/login.aspx?direct=true&db=edsclk&an=edsclk.15667892&scope=site

[6] HUANG, B. Q., Y. B. ZHANG a M. T. KECHADI, 2007. Preprocessing Techniques for Online Handwriting Recognition. *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)* [online]. IEEE, **754**, 793-800 [cit. 2020-08-02]. DOI: 10.1109/ISDA.2007.31. ISBN 0-7695-2976-3. ISSN 1742-6588. Dostupné z: http://ieeexplore.ieee.org/document/4389705/

[7] GUARAV, Kumar a Pradeep BHATIA, 2013. *Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition* [online]. 14-22 [cit. 2020-08-02]. DOI: 10.13140/RG.2.1.3896.7. Dostupné z: https://www.researchgate.net/publication/255707843_Analytical_Review_of_Preprocessing_Techniques_for_Offline_Handwritten_Character_Recognition

[8] ALGINAHI, Yasser, 2010. Preprocessing Techniques in Character Recognition. *Character Recognition* [online]. Sciyo [cit. 2020-08-03]. DOI: 10.5772/9776. ISBN 978-953-307-105-3. Dostupné z: http://www.intechopen.com/books/character-recognition/preprocessing-techniques-in-character-recognition

[9] KIM, Gyeonghwan, Venu GOVINDARAJU a Sargur N. SRIHARI, 1999. An architecture for handwritten text recognition systems. *International Journal on Document Analysis and Recognition* [online]. **2**(1), 37-44 [cit. 2020-08-03]. DOI: 10.1007/s100320050035. ISSN 1433-2833. Dostupné z: http://link.springer.com/10.1007/s100320050035

[10] DOERMANN, David a Stefan JAEGER, 2008. *Arabic and Chinese Handwriting Recognition*. Nemecko: Springer, s. 116-118. ISBN 978-3-540-78198-1.

[11] DAVE, Namrata, 2015. Segmentation Methods for Hand Written Character Recognition. *International Journal of Signal Processing, Image Processing and Pattern Recognition* [online]. **8**(4), 155-164 [cit. 2020-08-03]. DOI: 10.14257/ijsip.2015.8.4.14. ISSN 20054254. Dostupné z: http://article.nadiapub.com/IJSIP/vol8_no4/14.pdf

[12] ALAEI, Alireza, P. NAGABHUSHAN a Umapada PAL, 2011. Piece-wise painting technique for line segmentation of unconstrained handwritten text: a specific study with Persian text documents. *Pattern Analysis and Applications* [online]. **14**(4), 381-394 [cit. 2020-08-03]. DOI: 10.1007/s10044-011-0226-x. ISSN 1433-7541. Dostupné z: http://link.springer.com/10.1007/s10044-011-0226-x

[13] KAUR, Amandeep, Seema BAGHLA a Sunil KUMAR, 2015. Study Of Various Character Segmentation Techniques For Handwritten Off-Line Cursive Words: A Review. *International Journal of Advances in Science Engineering and Technolog* [online]. **3**(3), 154-158 [cit. 2020-08-02]. ISSN 2321-9009. Dostupné z: http://www.iraj.in/journal/journal_file/journal_pdf/6-162-1440573382154-158.pdf

[14] CHOUDHARY, Amit, 2014. A Review of Various Character Segmentation Techniques for Cursive Handwritten Words Recognition. *International Journal of Information & Computation Technology* [online]. **4**(6), 559-564 [cit. 2020-08-02]. ISSN 0974-2239. Dostupné z: https://www.ripublication.com/irph/ijict_spl/ijictv4n6spl_02.pdf

[15] MANDAL, Durbadal et al., 2015. *Intelligent Computing and Applications: Proceedings of the International Conference on ICA, 22-24 December 2014*. Springer. ISBN 8132222679.

[16] BHATIA, Sanjiv et al., 2018. *Advances in Computer Communication and Computational Sciences*. 2. Springer. ISBN 978-981-1303-432.

[17] PRASAD, Vijay a Y. Jayanta SINGH, 2013. A study on method of feature extraction for Handwritten Character Recognition. *Indian Journal of Science and Technolog* [online]. **6**, 174-178 [cit. 2020-08-02]. Dostupné z: https://www.researchgate.net/publication/258029618_A_study_on_method_of_feature_extraction_for_Handwritten_Character_Recognition

[18] PURKAIT, Pulak. *Optical Handwritten Character/Numeral Recognition* [online]. In: . [cit. 2020-08-02]. Dostupné z: https://www.isical.ac.in/~vlrg/sites/default/files/Pulak/Off-Line%20Handwritten%20OCR.pdf

[19] KEISHAM, Kanchan a Sunanda DIXIT, 2015. Techniques for Text,Line and Word Segmentation. *International Journal of Engineering Research & Technology (IJERT)* [online]. **3**(19) [cit. 2020-08-02]. ISSN 2278-0181. Dostupné z: https://www.ijert.org/research/techniques-for-text-line-and-word-segmentation-IJERTCONV3IS19065.pdf

[20] ZHANG, Harry, 2004. *The Optimality of Naive Bayes* [online]. [cit. 2020-08-02]. Dostupné z: http://www.cs.unb.ca/~hzhang/publications/FLAIRS04ZhangH.pdf

[21] DE STEFANO, C., F. FONTANELLA a A. SCOTTO DI FRECA, 2012. A Novel Naive Bayes Voting Strategy for Combining Classifiers. *2012 International Conference on Frontiers in Handwriting Recognition* [online]. IEEE, 467-472 [cit. 2020-08-03]. DOI: 10.1109/ICFHR.2012.166. ISBN 978-1-4673-2262-1. Dostupné z: http://ieeexplore.ieee.org/document/6424439/

[22] DHAKA, Vijaypal Singh, Manoj KUMAR a Prashant CHAUDHARY, 2015. *Offline Handwritten English Script Recognition: A Survey* [online]. 114-124 [cit. 2020-08-02]. Dostupné z: https://www.ijana.in/Special%20Issue/C20.pdf

[23] LECUN, Yann, Corinna CORTES a Christopher J.C. BURGES. *THE MNIST DATABASE of handwritten digits* [online]. [cit. 2020-08-02]. Dostupné z: http://yann.lecun.com/exdb/mnist/

[24] MOHAPATRA, Ramesh Kumar, Banshidhar MAJHI a Sanjay Kumar JENA, 2015. Classification performance analysis of MNIST Dataset utilizing a Multi-resolution Technique. *2015 International Conference on Computing, Communication and Security (ICCCS)* [online]. IEEE, 1-5 [cit. 2020-08-03]. DOI: 10.1109/CCCS.2015.7374136. ISBN 978-1-4673-9354-6. Dostupné z: http://ieeexplore.ieee.org/document/7374136/

[25] COHEN, Gregory et al., 2017. *EMNIST: an extension of MNIST to handwritten letters* [online]. [cit. 2020-08-02]. Dostupné z: https://arxiv.org/abs/1702.05373

[26] COHEN, Gregory et al., 2017. EMNIST: Extending MNIST to handwritten letters. *2017 International Joint Conference on Neural Networks (IJCNN)* [online]. IEEE, 2921-2926 [cit. 2020-08-03]. DOI: 10.1109/IJCNN.2017.7966217. ISBN 978-1-5090-6182-2. Dostupné z: http://ieeexplore.ieee.org/document/7966217/

[27] PALATNIK DE SOUSA, Iam, 2018. Convolutional ensembles for Arabic Handwritten Character and Digit Recognition. *PeerJ Computer Science* [online]. **4** [cit. 2020-08-03]. DOI: 10.7717/peerj-cs.167. ISSN 2376-5992. Dostupné z: https://peerj.com/articles/cs-167

[28] EL-SAWY, Ahmed, Hazem EL-BAKRY a Mohamed LOEY, 2017. CNN for Handwritten Arabic Digits Recognition Based on LeNet-5. *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016*. Springer, s. 566-575. ISBN 978-3-319-48307-8.

[29] EL-SAWY, Ahmed, Mohamed LOEY a Hazem M EL-BAKRY, 2017. Arabic Handwritten Characters Recognition using ConvolutionalNeural Network. *WSEAS Transactions on Computer Research* [online]. **5**, 11-19 [cit. 2020-08-02]. ISSN 2415-1513. Dostupné z: https://www.researchgate.net/publication/313891953_Arabic_Handwritten_Characters_Recognition_using_Convolutional_Neural_Network

[30] ROY, K. et al., 2005. Oriya handwritten numeral recognition system. *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)* [online]. IEEE, 770-774 Vol. 2 [cit. 2020-08-03]. DOI: 10.1109/ICDAR.2005.183. ISBN 0-7695-2420-6. Dostupné z: http://ieeexplore.ieee.org/document/1575649/

[31] ALAM, Samiul et al., 2018. *NumtaDB - Assembled Bengali Handwritten Digits* [online]. [cit. 2020-08-03]. Dostupné z: https://arxiv.org/abs/1806.02452

[32] BISWAS, Mithun et al., 2017. *BanglaLekha-Isolated: A Comprehensive Bangla Handwritten Character Dataset* [online]. [cit. 2020-08-03]. Dostupné z: https://arxiv.org/abs/1703.10661

[33] SHARIF, S.M.A. a Mahdin MAHBOOB, 2017. Evil method: A deep CNN model for Bangla handwritten numeral classification. *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)* [online]. IEEE, 217-222 [cit. 2020-08-03]. DOI: 10.1109/ICAEE.2017.8255356. ISBN 978-1-5386-0869-2. Dostupné z: http://ieeexplore.ieee.org/document/8255356/

[34] ACHARYA, Shailesh, Ashok Kumar PANT a Prashnna Kumar GYAWALI, 2015. Deep learning based large scale handwritten Devanagari character recognition. *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)* [online]. IEEE, 1-6 [cit. 2020-08-03]. DOI: 10.1109/SKIMA.2015.7400041. ISBN 978-1-4673-6744-8. Dostupné z: http://ieeexplore.ieee.org/document/7400041/

[35] BAG, SOUMEN a GAURAV HARIT, 2013. A survey on optical character recognition for Bangla and Devanagari scripts. *Sadhana* [online]. **38**(1), 133-168 [cit. 2020-08-03]. DOI: 10.1007/s12046-013-0121-9. ISSN 0256-2499. Dostupné z: http://link.springer.com/10.1007/s12046-013-0121-9

[36] KUMAR, Munish, M. K. JINDAL a R. K. SHARMA, 2011. K-nearest neighbor based offline handwritten Gurmukhi character recognition. *2011 International Conference on Image Information Processing* [online]. IEEE, 1-4 [cit. 2020-08-03]. DOI: 10.1109/ICIIP.2011.6108863. ISBN 978-1-61284-861-7. Dostupné z: http://ieeexplore.ieee.org/document/6108863/

[37] BALDOMINOS, Alejandro, Yago SAEZ a Pedro ISASI, 2019. A Survey of Handwritten Character Recognition with MNIST and EMNIST. *Applied Sciences* [online]. **9**(15) [cit. 2020-08-03]. DOI: 10.3390/app9153169. ISSN 2076-3417. Dostupné z: https://www.mdpi.com/2076-3417/9/15/3169

[38] KAZIHA, Omar a Talal BONNY, 2019. A Comparison of Quantized Convolutional and LSTM Recurrent Neural Network Models Using MNIST. *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)* [online]. IEEE, 1-5 [cit. 2020-08-03]. DOI: 10.1109/ICECTA48151.2019.8959793. ISBN 978-1-7281-5532-6. Dostupné z: https://ieeexplore.ieee.org/document/8959793/

[39] TEOW, Matthew Y. W., 2017. Understanding convolutional neural networks using a minimal model for handwritten digit recognition. *2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS)* [online]. IEEE, 167-172 [cit. 2020-08-03]. DOI: 10.1109/I2CACIS.2017.8239052. ISBN 978-1-5386-0846-3. Dostupné z: http://ieeexplore.ieee.org/document/8239052/

[40] GHADEKAR, Preman, Shubham INGOLE a Dhruv SONONE, 2018. Handwritten Digit and Letter Recognition Using Hybrid DWT-DCT with KNN and SVM Classifier. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)* [online]. IEEE, 1-6 [cit. 2020-08-03]. DOI: 10.1109/ICCUBEA.2018.8697684. ISBN 978-1-5386-5257-2. Dostupné z: https://ieeexplore.ieee.org/document/8697684/

[41] ROBERTS, Jozette V. a Isaac DIALSINGH, 2016. On the Performance of Classification Techniques with Pixel Removal Applied to Digit Recognition. *International Journal of Intelligent Systems and Applications* [online]. **8**(8), 38-47 [cit. 2020-08-03]. DOI: 10.5815/ijisa.2016.08.05. ISSN 2074904X. Dostupné z: http://www.mecs-press.org/ijisa/ijisa-v8-n8/v8n8-5.html

[42] BAKAS, Jamimamul, Mahabub H. a Ayatullah F., 2016. A Comparative Study of Various Classifiers for Character Recognition on Multi-script Databases. *International Journal of Computer Applications* [online]. **155**(3), 1-5 [cit. 2020-08-03]. DOI: 10.5120/ijca2016912277. ISSN 09758887. Dostupné z: http://www.ijcaonline.org/archives/volume155/number3/bakas-2016-ijca-912277.pdf

[43] AHAMED, Pratik et al. Handwritten Arabic numerals recognition using convolutional neural network. *Journal of Ambient Intelligence and Humanized Computing* [online]. [cit. 2020-08-03]. DOI: 10.1007/s12652-020-01901-7. ISSN 1868-5137. Dostupné z: http://link.springer.com/10.1007/s12652-020-01901-7

[44] ALI, Amani Ali Ahmed a Suresha M, 2019. Arabic Handwritten Character Recognition Using Machine Learning Approaches. *2019 Fifth International Conference on Image Information Processing (ICIIP)* [online]. IEEE, 187-192 [cit. 2020-08-03]. DOI: 10.1109/ICIIP47207.2019.8985839. ISBN 978-1-7281-0899-5. Dostupné z: https://ieeexplore.ieee.org/document/8985839/

[45] H. AKHAND, M. A. a Mahtab AHMED, 2016. Convolutional Neural Network based Handwritten Bengali and Bengali-English Mixed Numeral Recognition. *International Journal of Image, Graphics and Signal Processing* [online]. **8**(9), 40-50 [cit. 2020-08-03]. DOI: 10.5815/ijigsp.2016.09.06. ISSN 20749074. Dostupné z: http://www.mecs-press.org/ijigsp/ijigsp-v8-n9/v8n9-6.html

[46] AHMED, Mahtab, M. A. H. AKHAND a M. M. HAFIZUR RAHMAN, 2019. Recognizing Bangla Handwritten Numeral Utilizing Deep Long Short Term Memory. *International Journal of Image, Graphics and Signal Processing* [online]. **11**(1), 23-32 [cit. 2020-08-03]. DOI: 10.5815/ijigsp.2019.01.03. ISSN 20749074. Dostupné z: http://www.mecs-press.org/ijigsp/ijigsp-v11-n1/v11n1-3.html

[47] RAHMAN, Md. Mahbubar et al., 2015. Bangla Handwritten Character Recognition using Convolutional Neural Network. *International Journal of Image, Graphics and Signal Processing* [online]. **7**(8), 42-49 [cit. 2020-08-03]. DOI: 10.5815/ijigsp.2015.08.05. ISSN 20749074. Dostupné z: http://www.mecs-press.org/ijigsp/ijigsp-v7-n8/v7n8-5.html

[48] CHOWDHURY, Rumman Rashid et al., 2019. *Bangla Handwritten Character Recognition using Convolutional Neural Network with Data Augmentation* [online]. IEEE, 318-323 [cit. 2020-08-03]. DOI: 10.1109/ICIEV.2019.8858545. ISBN 978-1-7281-0786-8. Dostupné z: https://ieeexplore.ieee.org/document/8858545/

[49] DAS, Nibaran et al., 2015. Handwritten Bangla character recognition using a soft computing paradigm embedded in two pass approach. *Pattern Recognition* [online]. **48**(6), 2054-2071 [cit. 2020-08-03]. DOI: 10.1016/j.patcog.2014.12.011. ISSN 00313203. Dostupné z: https://linkinghub.elsevier.com/retrieve/pii/S0031320314005202

[50] SONKUSARE, Manoj, Roopam GUPTA a Asmita MOGHE, 2019. *A Reviewon Handwritten Devanagari CharacterRecognition* [online]. [cit. 2020-08-03]. Dostupné z: https://easychair.org/publications/preprint/GhHF

[51] KUMAR, Munish, M. K. JINDAL a R. K. SHARMA, 2011. K-nearest neighbor based offline handwritten Gurmukhi character recognition. *2011 International Conference on Image Information Processing* [online]. IEEE, 1-4 [cit. 2020-08-03]. DOI: 10.1109/ICIIP.2011.6108863. ISBN 978-1-61284-861-7. Dostupné z: http://ieeexplore.ieee.org/document/6108863/

[52] SINGH, Gurpreet a Manoj SACHAN, 2014. Multi-layer perceptron (MLP) neural network technique for offline handwritten Gurmukhi character recognition. *2014 IEEE International Conference on Computational Intelligence and Computing Research* [online]. IEEE, 1-5 [cit. 2020-08-03]. DOI: 10.1109/ICCIC.2014.7238334. ISBN 978-1-4799-3974-9. Dostupné z: http://ieeexplore.ieee.org/document/7238334/

[53] KUMAR, Munish, Rishav Kumar SHARMA a M. K. JINDAL, 2011. *SVM Based Offline Handwritten Gurmukhi Character Recognition* [online]. [cit. 2020-08-03]. Dostupné z: https://www.researchgate.net/publication/268514406_SVM_based_Offline_Handwritten_Gurmukhi_Character_Recognition

[54] GUPTA, Sheifali et al., 2019. Offline Handwritten Gurumukhi Character Recognition System Using Convolution Neural Network. *Journal of Computational and Theoretical Nanoscience* [online]. **16**(10), 4164-4169 [cit. 2020-08-03]. DOI: 10.1166/jctn.2019.8497. ISSN 1546-1955. Dostupné z: https://www.ingentaconnect.com/content/10.1166/jctn.2019.8497

[55] MEYER, Bradley, Marwan BIKDASH a Xiangfeng DAI, 2017. Fine-grained financial news sentiment analysis. *SoutheastCon 2017* [online]. IEEE, 1-8 [cit. 2020-08-03]. DOI: 10.1109/SECON.2017.7925378. ISBN 978-1-5386-1539-3. Dostupné z: https://ieeexplore.ieee.org/document/7925378/

[56] CHEN, Shengfeng et al., 2018. *Offline Handwritten Digits Recognition Using Machine learning* [online]. 274-286 [cit. 2020-08-03]. Dostupné z: http://ieomsociety.org/dc2018/papers/123.pdf

[57] SLAVIK, P. a V. GOVINDARAJU. Equivalence of different methods for slant and skew corrections in word recognition applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online]. **23**(3), 323-326 [cit. 2020-08-03]. DOI: 10.1109/34.910885. ISSN 01628828. Dostupné z: http://ieeexplore.ieee.org/document/910885/

[58] LEE, Hong a Brijesh VERMA, 2012. Binary segmentation algorithm for English cursive handwriting recognition. *Pattern Recognition* [online]. **45**(4), 1306-1317 [cit. 2020-08-03]. DOI: 10.1016/j.patcog.2011.09.015. ISSN 00313203. Dostupné z: https://linkinghub.elsevier.com/retrieve/pii/S0031320311003943

[59] WASEEM, Mohammad, 2020. How To Implement Classification In Machine Learning? *Edureka* [online]. [cit. 2020-08-03]. Dostupné z: https://www.edureka.co/blog/classification-in-machine-learning/?fbclid=IwAR0D8PHhxjneyiZlf2KbeWLrKQp-Auq65jeGlDFM4Ny2fxmIdsudn9blqAU

[60] GONFALONIERI, Alexandre, 2019. How to Build A Data Set For Your Machine Learning Project. *Towards data science* [online]. [cit. 2020-07-27]. Dostupné z: https://towardsdatascience.com/how-to-build-a-data-set-for-your-machine-learning-project-5b3b871881ac

[61] AL-SHATNAW, Atallah Mahmoud a Khairuddin OMAR, 2009. Skew Detection and Correction Technique for Arabic Document Images Based on Centre of Gravity. *Journal of Computer Science* [online]. **5**(5), 363-368 [cit. 2020-08-03]. DOI: 10.3844/jcssp.2009.363.368. ISSN 15493636. Dostupné z: http://www.thescipub.com/abstract/?doi=jcssp.2009.363.368

[62] HULL, JONATHAN J., 1998. DOCUMENT IMAGE SKEW DETECTION: SURVEY AND ANNOTATED BIBLIOGRAPHY. *Document Analysis Systems II* [online]. WORLD SCIENTIFIC, 40-64 [cit. 2020-08-03]. Series in Machine Perception and Artificial Intelligence. DOI: 10.1142/9789812797704_0003. ISBN 978-981-02-3103-3. Dostupné z: http://www.worldscientific.com/doi/abs/10.1142/9789812797704_0003

[63] DAS, Nibaran et al., 2010. Handwritten Bangla Basic and Compound character recognition using MLP and SVM classifier. *Journal of Computing* [online]. **2**(2), 109-115 [cit. 2020-08-03]. ISSN 2151-9617. Dostupné z: https://arxiv.org/ftp/arxiv/papers/1002/1002.4040.pdf

[64] KUMAR, Ashwin Ram, 2020. Mnist_classification. GitHub: The world's leading software *development platform* [online]. [cit. 2020-08-03]. Dostupné z: https://github.com/AshRK1997/mnist_classification

[65] DUTT, Anuj a Banee ISHAQUE K, 2020. Handwritten-Digit-Recognition-using-Deep-Learning. *GitHub: The world's leading software development platform* [online]. [cit. 2020-08-03]. Dostupné z: https://github.com/anujdutt9/Handwritten-Digit-Recognition-using-Deep-Learning/commits/master

[66] KHATTAK, Saad, 2020. RNN and LSTM for Image Classifications. *GitHub: The world's leading software development platform* [online]. [cit. 2020-08-03]. Dostupné z: https://github.com/saadkiet/RNN-and-LSTM-for-Image-Classifications

[67] NAIDU, Amilineni Roshni, 2020. MLP-for-MNIST-dataset. *GitHub: The world's leading software development platform* [online]. [cit. 2020-08-03]. Dostupné z: https://github.com/roshninaidu31/MLP-for-MNIST-dataset

ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK

NIST	National Institute of Standards and Technology
MNIST	Mixed NIST
EMNIST	Extended MNIST
BHDDB	Bengali Handwritten Digits Database
CMATER	Center for Microprocessor Application for Training Education and Research
SD-1	Special Database 1
SD-3	Special Database 3
k-NN	k-Nearest Neighbors
SVM	Support Vector Machines
MLP	Multi-layer Perceptron
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
NB	Naive Bayes
RBF	Radial Basis Function
PSC	Potencial Segmentation Column
PNG	Portable Network Graphics
ISI	Indian Statistical Institute
RAM	Random Access Memory
IRF	Impulse Response Function

ZOZNAM OBRÁZKOV

Obrázok 1: Princíp fungovania k-NN [1] 13
Obrázok 2: Rozhodovacie plochy [55] 14
Obrázok 3: Klasifikácia podporných vektorov: (a) lineárne oddeliteľné dáta, (b) lineárne neoddeliteľné dáta [1] 15
Obrázok 4: Kernel mapping: nelineárne oddeliteľné dáta a lineárne oddeliteľné dáta [2] .. 15
Obrázok 5: Príklad viacvrstového perceptrónu [3] 16
Obrázok 6: Konvolučná vrstva [4] 18
Obrázok 7: Subsamplingová (pooling) vrstva [4] 18
Obrázok 8: Konvolučná neurónová sieť [1] 19
Obrázok 9: Vizualizácia pamäťovej bunky [1] 21
Obrázok 10: Nespracovaný obrázok (vľavo) a obrázok po použití Mediánovej filtrácie (vpravo) [56] 23
Obrázok 11: Ručne písané slovo pred a po korekcii zošikmenia [57] 24
Obrázok 12: Segmentácia riadkov [12] 26
Obrázok 13: Správna segmentácia slova "Buchanan" [58] 27
Obrázok 14: Vzorky slov po segmentácii [13] 28
Obrázok 15: Implicitná segmentácia [13] 29
Obrázok 16: Obrázok pred Zoningom a po Zoningu [17] 32
Obrázok 17: Štatistická metóda: Projekčný histogram [17] 32
Obrázok 18: Štatistická metóda: Profily [17] 33
Obrázok 19: Kríženia a vzdialenosti [17] 33
Obrázok 20: Štrukturálna metóda [17] 34
Obrázok 21: 100 ukážkových obrázkov číslic z datasetu MNIST [24] 37
Obrázok 22: Vzorky všetkých písmen a číslic v súbore údajov EMNIST [37] 38
Obrázok 23: Príklady Oriya ručne písaných číslic [30] 40
Obrázok 24: Modifikované znaky samohlásky [35] 41
Obrázok 25: Presnosť klasifikátora CNN 47
Obrázok 26: Presnosť klasifikátora LSTM 48
Obrázok 27: Úspešnosť klasifikátora MLP 49
Obrázok 28: Graf porovnaní úspešností klasifikátorov 51

ZOZNAM TABULIEK

Tabuľka 1: Porovnanie rôznych metód segmentácie [19], kvôli pôvodnosti a dohľadateľnosti nie sú metódy preložené do slovenského jazyka 30
Tabuľka 2: Porovnanie klasifikátorov na rôznych datasetoch 36
Tabuľka 3: Výsledky testovaných aplikácií 45
Tabuľka 4: Konfúzna matica pre k-NN MNIST 46
Tabuľka 5: Konfúzna matica pre SVM MNIST 46
Tabuľka 6: Konfúzna matica pre CNN MNIST 48
Tabuľka 7: Konfúzna matica pre LSTM MNIST 49
Tabuľka 8: Konfúzna matica pre MLP MNIST 50

ZOZNAM PRÍLOH

Príloha P I: CD s bakalárskou prácou vrátane zdrojových kódov testovaných aplikácií

PRÍLOHA P I: CD

Priložené CD obsahuje:

- Text
    - BP_AndrejOvecka_A16439.pdf
- Zdrojové kódy
    - prilohy.zip – zdrojové kódy testovaných aplikácií